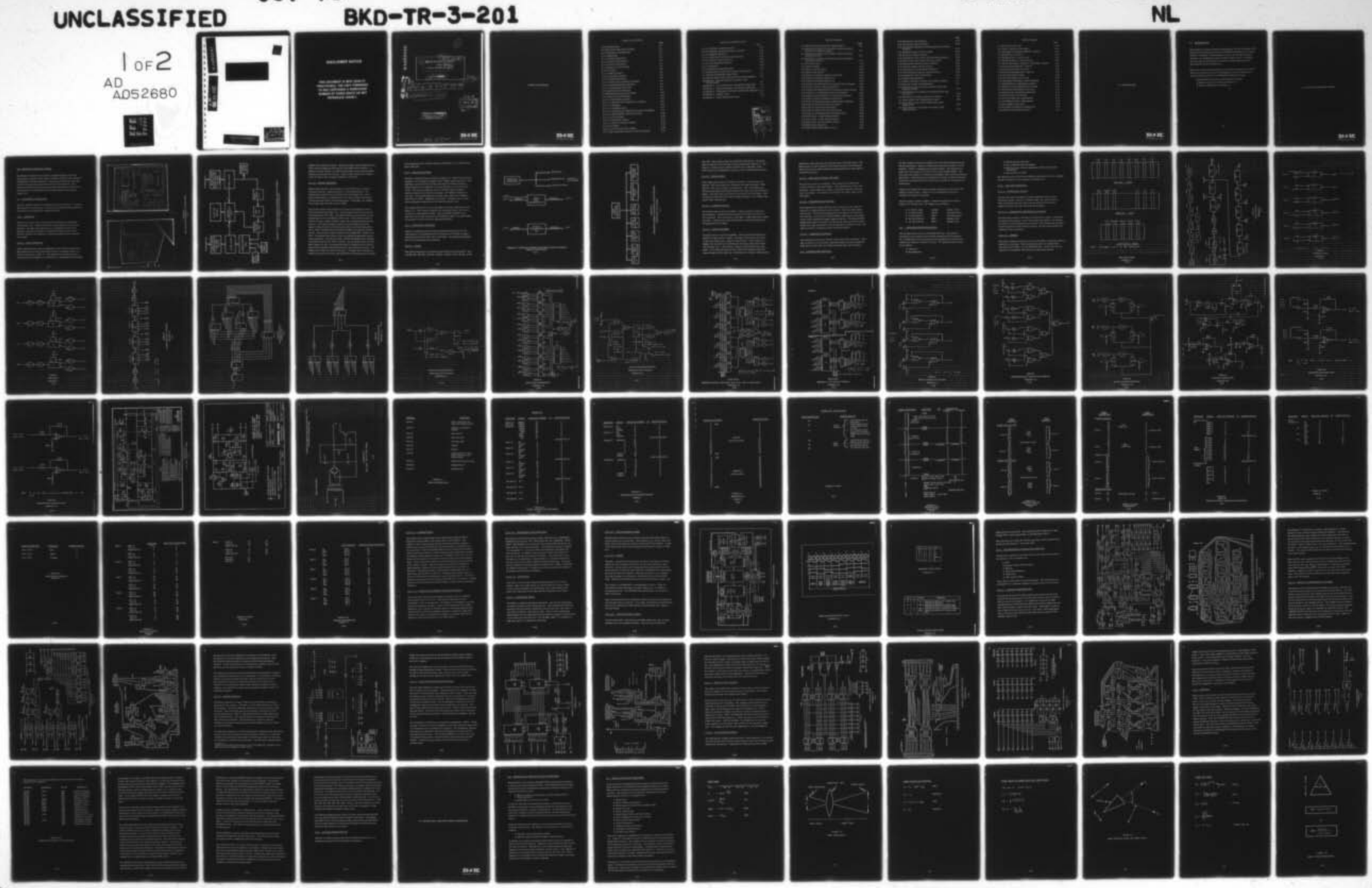
AD-A052 680 B-K DYNAMICS INC ROCKVILLE MD F/G 9/2
QUARTERLY INTERIM TECHNICAL REPORT (4TH), CONTRACT DAAH01-75-C---ETC(U)
OCT 75 DAAH01-75-C-0194
UNCLASSIFIED BKD-TR-3-201 NL

1 OF 2
AD
A052680



AD A052680

(1)

(14) BKD- TR-3-201

(6) [redacted] Quarterly Interim Technical Report (4th), Contract DAAHØ1-75-C-Ø194.

(11) 15 Oct[redacted]75

(9) Rept. for 15 Jul-15 Oct 75.

(12) 171p.

(15) DAAHØ1-75-C-Ø194

287 575

B&K DYNAMICS, INC.

TABLE OF CONTENTS

B K
DYNAMICS, INC.

TABLE OF CONTENTS

TABLE OF CONTENTS CONT.

LIST OF FIGURES

Page

LIST OF TABLES

1.0 INTRODUCTION

1.0 INTRODUCTION

B-K Dynamics' activities during the fourth quarter (15 July-15 October 1975) focused on documentation of the Interim STINGER simulation and its related hardware components. Documentation on each of the computer system's interfaces was completed and delivered under separate cover. Additional time was spent on documenting the equations actually implemented and describing the system configuration.

This report presents documentation of the Software Checkout Chassis (SCC) which was constructed to assist in debugging the real time software and provides additional details on the STINGER simulation including;

- A functional description of the simulation elements,
- Specific equations currently implemented, and
- Detailed input/output descriptions.

2.0 SOFTWARE CHECKOUT CHASSIS

2.0 SOFTWARE CHECKOUT CHASSIS

The Software Checkout Chassis (SCC) is a general purpose test device constructed to perform several specific software checkout tasks associated with MICOM's STINGER simulation. The tasks performed are described in BKD's 3rd Quarterly Report (TR-3-197) dated 22 July 1975. This section documents the SCC design and presents information necessary for the potential user to understand its functional capabilities for general purpose use.

2.1 FUNCTIONAL DESCRIPTION

The SCC, shown in Figure 2.1, performs three basic functions: it receives and transmits 5 volt discretes, it receives and transmits analog signals and it provides general purpose test support functions.

2.1.1 DISCRETES

The SCC can receive and transmit 32 discrete signals and display their current logic level. Figure 2.2 presents a block diagram of the discrete functions. An eight bit microprocessor is incorporated to provide an automatic sequencing capability for repetitive testing or for use where discrete responses are required in a time frame shorter than an operator can respond.

2.1.1.1 INPUT DISCRETES

Sixteen input discretes come in through patch holes on the front panel and go to logic select circuitry which allows the user to designate either a +5 volt or 0 volt level as a logic "1". The output of this circuitry goes to a 16 bit latch which is updated at a 250 KHZ rate or optionally can be manually

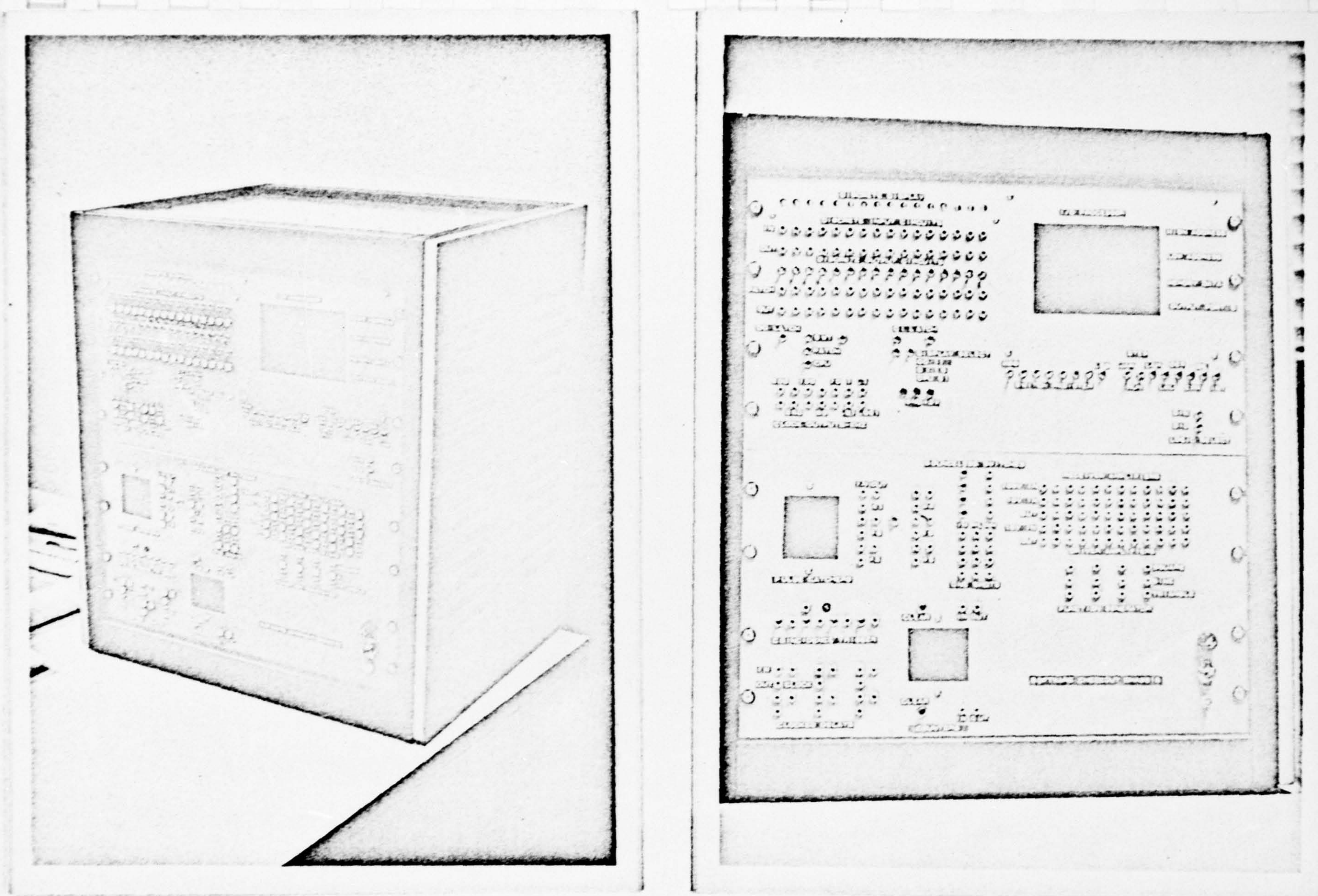


Figure 2.1
THE SOFTWARE CHECKOUT CHASSIS (SCC)

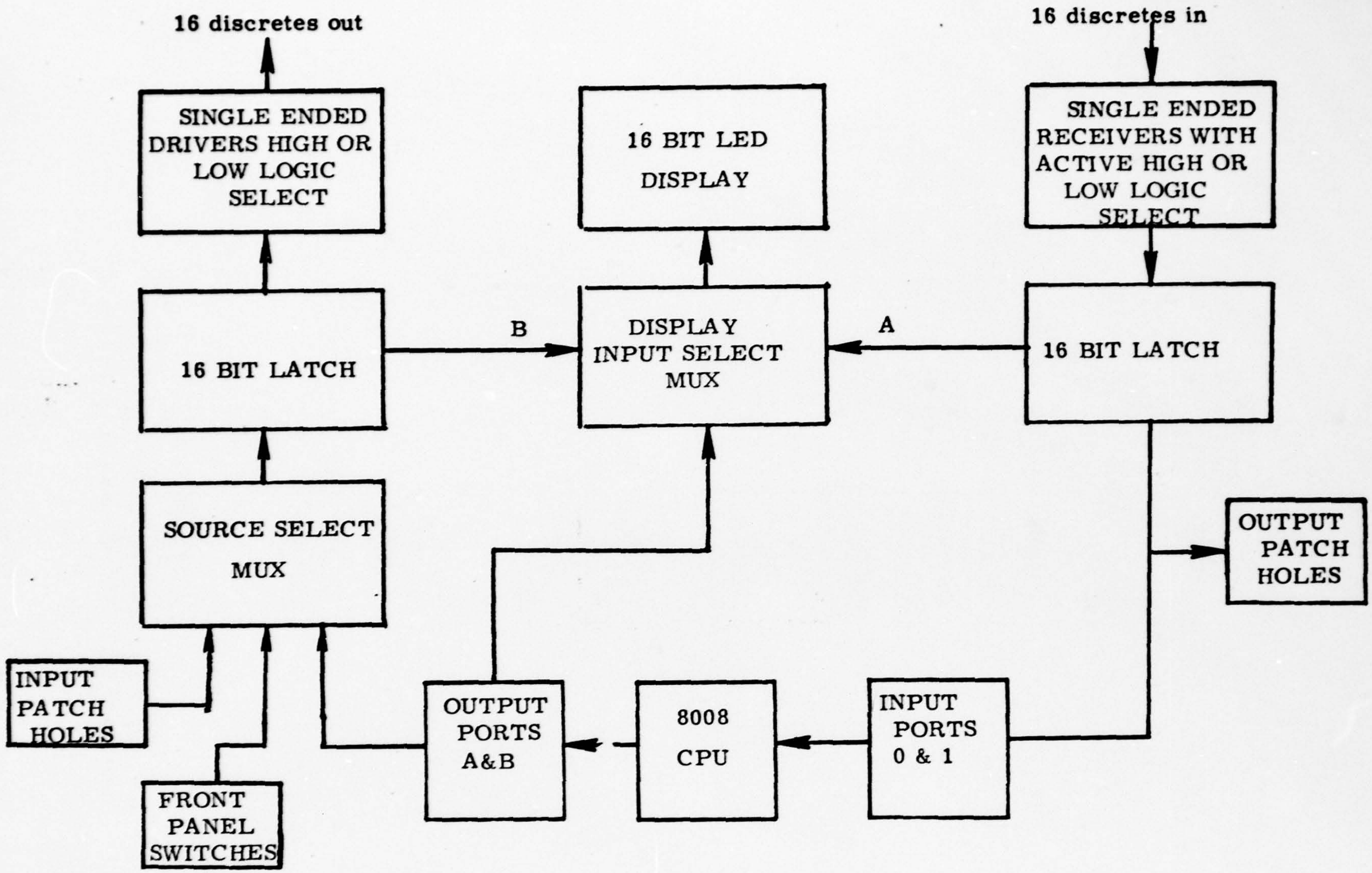


FIGURE 2.2 CHECKOUT CHASSIS FUNCTIONAL BLOCK DIAGRAM
(DISCRETE HANDLING AND DISPLAY)

updated with a pushbutton switch. The latch contents can be displayed on the 16 bit discrete display by selecting 10 on the display select switches. In addition the latch contents are routed to patch holes on the front panel where they may be patched to test devices. Also the latch contents are available to the microprocessor through its input ports, #0 and #1.

2.1.1.2 OUTPUT DISCRETES

Sixteen output discretes can be generated from either of three sources: front panel switches, input patch holes, or the microprocessor. The selected input (selection is by switches on the front panel) sets a 16 bit latch whose contents are conditioned for a +5 volt or 0 logic 1 and routed to front panel patch holes via DTL power gates. These gates are adequate for driving a fifty foot cable at up to 1 MHZ.

The discrete display and patch holes are located in the upper left corner of the front panel (see Figure 2.1). Discrete inputs from external sources are patched in the "IN" holes. The state of these discretes may be sensed by patching into the corresponding "OUT" hole. These terminals may be used for driving a counter, delay or single shot in the general purpose test circuitry. The discrete output controls are located below the discrete inputs. Sixteen switches and sixteen patch holes are provided along with a row of "OUT" terminals where the discrete outputs are routed to external devices. The DO source select switches (3) are labeled SWT, PATCH, and CPU. The operator must select a source by putting the appropriate switch in the up position. The DO and DI latch controls and display select switches are directly under the DO "OUT" terminal. For normal operations the latch select signals are left in the up position while for manual latch the switch is down and the momentary push button switch is used to latch the circuit. The display select codes are shown on the panel. The logic select switches are located on the right side of the front panel below the microprocessor controls.

In the up position these switches select inverted logic (i.e. +5 volts in will give a logic low).

2.1.2 ANALOG FUNCTIONS

The SCC's analog functions are shown in Figure 2.3. There are 12 receiver amplifiers, 12 transmitter amplifiers and a simple function generator. The amplifiers consist of 747 IC op amps with unity gain feedback networks. These amplifiers are capable of driving 50 foot lines at frequencies up to 2 KHZ. The receiver amplifiers have an additional input resistance network to permit receiving 100 volt signals. All amplifier outputs are $\pm$ 10 volts. The function generator provides sine, square and triangular wave forms at 20 HZ. Amplitude is $\pm$ 10 volts. Frequency can be varied only by adjusting trimmer pots on the function generator card.

The analog function I/O is on the front panel in the lower right corner (see Figure 2.1). Each receiver amplifier has three patch holes, 100 volts in, 10 volts in and the amplifier output. Each transmitter amplifier has a patch hole for 10 volts in and one for the amplifier output. Four patch holes are provided for each of the three functions generated.

2.1.3 AUXILLARY FUNCTIONS

Figure 2.4 outlines the seven auxillary functions performed by the SCC. The functions are implemented with TTL logic and are intended to support checkout activities with other TTL or DTL logic. The following paragraphs give a brief description of each function.

2.1.3.1 CLOCK

Nine clock frequencies are available at patch holes on the front panel. They are 500 KHZ, 250 KHZ, 100 KHZ, 50 KHZ, 10 KHZ, 1 KHZ, 100 HZ, 10 HZ,

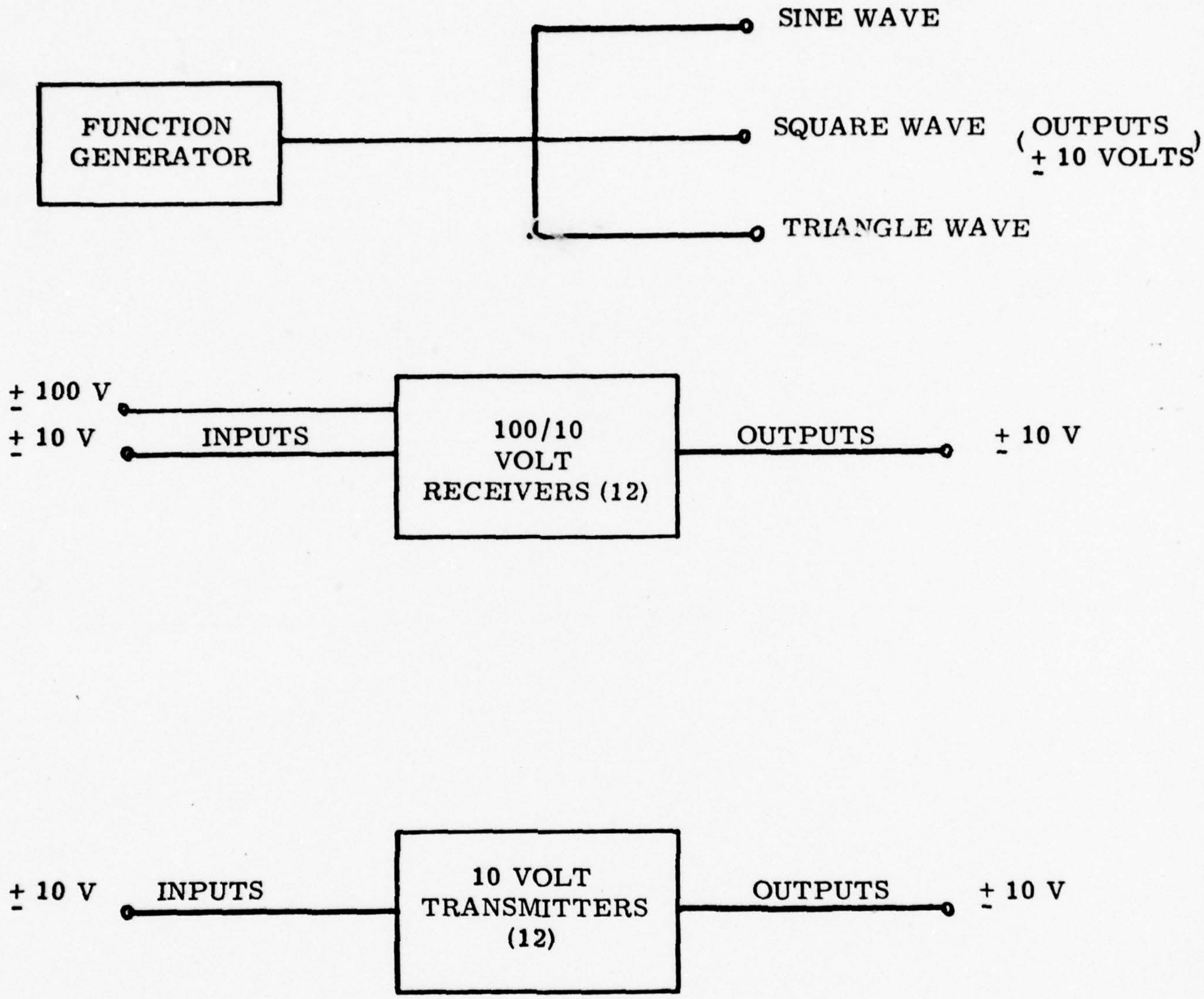


FIGURE 2.3 CHECKOUT CHASSIS FUNCTIONAL BLOCK DIAGRAM
(ANALOG FUNCTIONS)

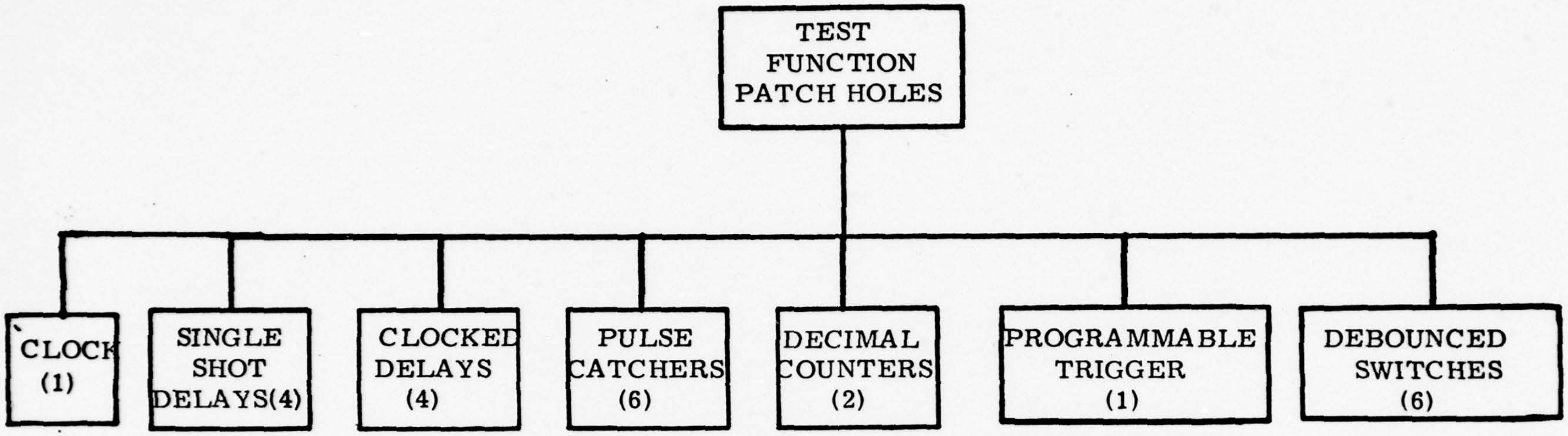


FIGURE 2.4

CHECKOUT CHASSIS FUNCTIONAL BLOCK DIAGRAM

(AUXILLARY FUNCTIONS)

and 1 HZ. Each signal is high for one-half the clock period. The patch holes are located on the left side of the front panel (see Figure 2.1). The clock is a 2 MHZ crystal oscillator divided by 4, to 400 KHZ, using a 74177 counter and subsequently divided by a chain of 7490 decade counters.

2.1.3.2 SINGLE SHOTS

Four single shot circuits are installed in the SCC. These circuits are basically 74123 dual single shots buffered at the input and output by 7400 NANDS. They fire on a high to low transition. Two of the outputs give 100 nanosecond pulses (blue patch holes) and the other two 500 manosecond pulses. The patch holes are located in the center of the front panel. Each circuit has an input patch hole and two output terminals, one normally high and the other normally low.

2.1.3.3 CLOCKED DELAYS

Six clocked delay circuits are installed. These circuits are 7491 8-bit shift registers. The patch holes are located in the lower left corner of the front panel. Each circuit has a clock input, a signal input and an output terminal. A delay of up to eight seconds may be achieved by patching in the appropriate clock signal.

2.1.3.4 PULSE CATCHERS

Six pulse catcher circuits are installed. These circuits use flip flops to indicate the logic level of a signal. A change in level causes the circuit to latch and indicate that a change in state occured. Each circuit has a display with two LED's one indicating that the input is high and the other that the input is low. The circuit is reset by a push button on the front panel. A low input will light the lower LED and a momentary high will cause the upper LED to light also, indicating that a change in state occured.

Pushing the clear button will show the final state of the input signal. The pulse catcher circuits terminate on the lower left side of the front panel. Each circuit has two input terminals to permit patching the input signal to an additional function.

2.1.3.5 FOUR DIGIT DECIMAL COUNTER

Two decimal counters are installed. They consist of (4) 7490 decade counters driving a 4 digit LED display. They increment on a high to low transition. Each counter has an input, an overflow output for cascading, and a clear button. The counters are located at the bottom center of the front panel.

2.1.3.6 PROGRAMMABLE TRIGGER

The programmable trigger allows the user to generate a level upon the occurance of a pattern entered on (4) switches. Up to four input signals may be used. The switch corresponding to each input is set to the level to be detected. If at any time all input signals simultaneously correspond to their switch settings the trigger output will go low. The programmable trigger patch holes are located at the bottom left corner of the front panel. Switches and input terminals for the four signals are located in a row. The output terminal is immediately above the input terminals.

2.1.3.7 DEBOUNCED SWITCHES

Four debounced switches are located in the center of the front panel. They may be used for firing one shots, generating output discretes through the patch holes and any other task requiring manual logic level switching.

2.2 CONSTRUCTION DRAWINGS

The SCC consists of one rack of digital cards, one rack of analog cards and the microprocessor. Figure 2.5 shows each rack and the number of cards installed. Table 2.1 designates each card's function. All construction is hand wired "wire-wrap" on 4.5" x 6.5" Vector cards (catalog #3662). The front panel (see Figure 2.1) contains all the display LED's and interconnection patch holes. Front panel to card connections are made via Molex connectors so that the front panel and/or card files can be removed for maintenance or modification.

Figures 2.6 through 2.22 contain schematic diagrams for each card in the analog and digital racks. Tables 2.2 to 2.10 designate all card and connector pin numbers by function.

The SCC contains 5 power supplies. Schematic diagrams are shown in Figures 2.23 through 2.25. The supplies are as follows:

• +5 Volt @ 8 amps	Lambda	(digital cards)
• +5 Volt @ 3 amps	Wortek	(microprocessor)
• -9 Volt @ 200 ma	EICO	(microprocessor)
• +14 Volt @ 150 ma	EICO	(analog cards)
• -14 Volt @ 150 ma	EICO	(analog cards)

2.3. MICROPROCESSOR DESCRIPTION

The microprocessor used is an Intel 8008 CPU chip. The system is constructed on five printed circuit boards manufactured by Techniques Inc. of Englewood, New Jersey and one wire wrapped circuit board which contains the computer's memory circuitry. The machine characteristics are as follows:

- 8 Bit word
- 48 Instructions

- 20 Microsecond cycle time
- TTL compatible input and outputs
- 256 Word static memory (expandable to 16K words directly addressable)
- One hardware interrupt

The following sections describe the 8008 microprocessor chip, the computer system components and the computer's instruction set.

2.3.1 8008 CHIP DESCRIPTION

2.3.1.1 FUNCTIONAL BLOCKS

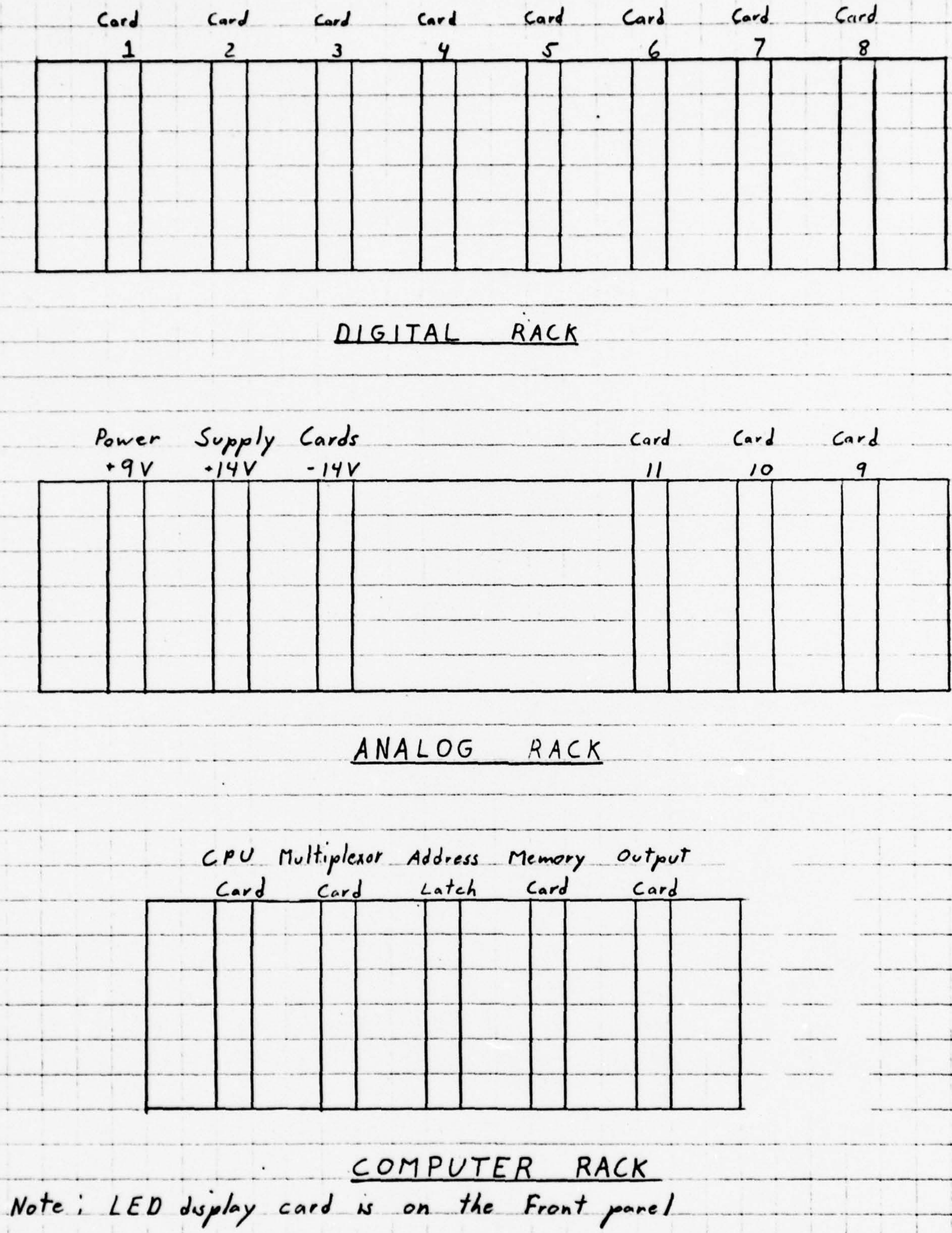
The four basic functional blocks of the Intel 8008 processor are the instruction register, memory, arithmetic logic unit, and I/O buffers. They communicate with each other over the internal 8-bit data bus. Figure 2.26 shows the chip's functional block diagram.

2.3.1.1.1 INSTRUCTION REGISTER AND CONTROL

The instruction register is the heart of all processor control. Instructions are fetched from memory, stored in the instruction register, and decoded for control of both the memories and the ALU. Since instruction executions do not all require the same number of states, the instruction decoder also controls the state transitions.

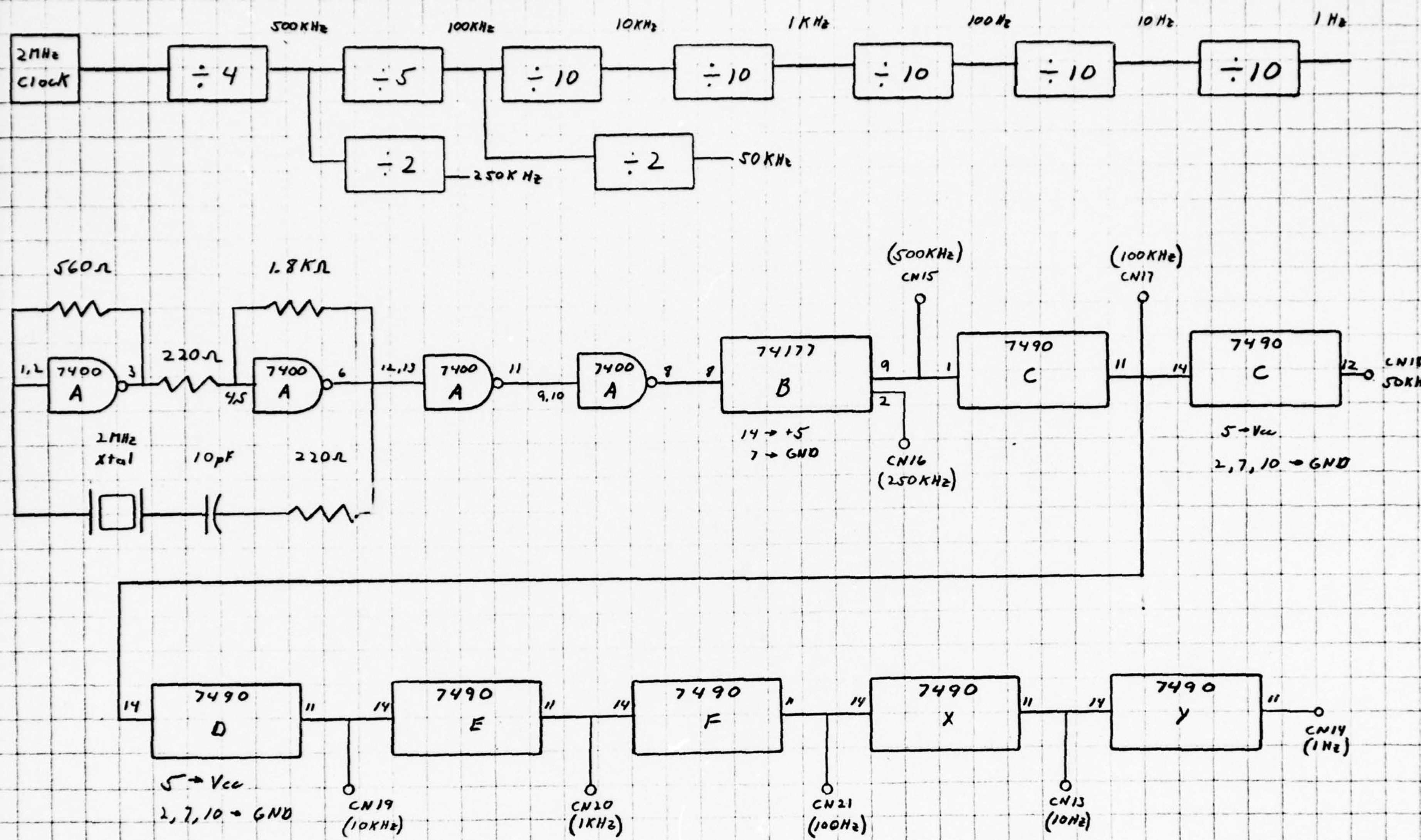
2.3.1.1.2 MEMORY

Two separate dynamic memories are used in the 8008, the pushdown address stack and a scratch pad. These internal memories are automatically refreshed by each WAIT, T3, and STOPPED state. In the worst case the memories are completely refreshed every eighty clock periods.



SCC CARD FILES

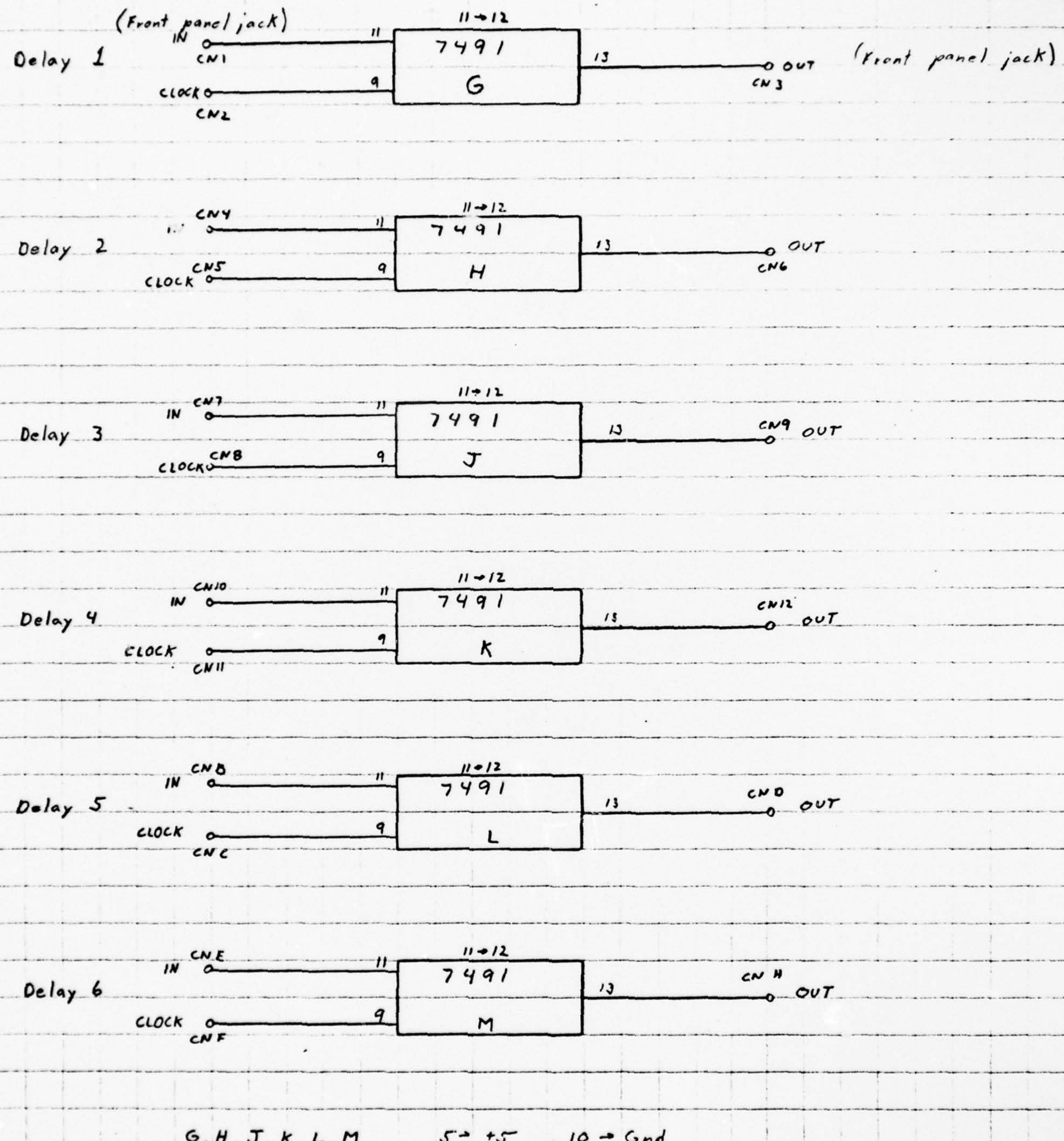
FIGURE 2.5



CARD #1

CLOCK CIRCUIT

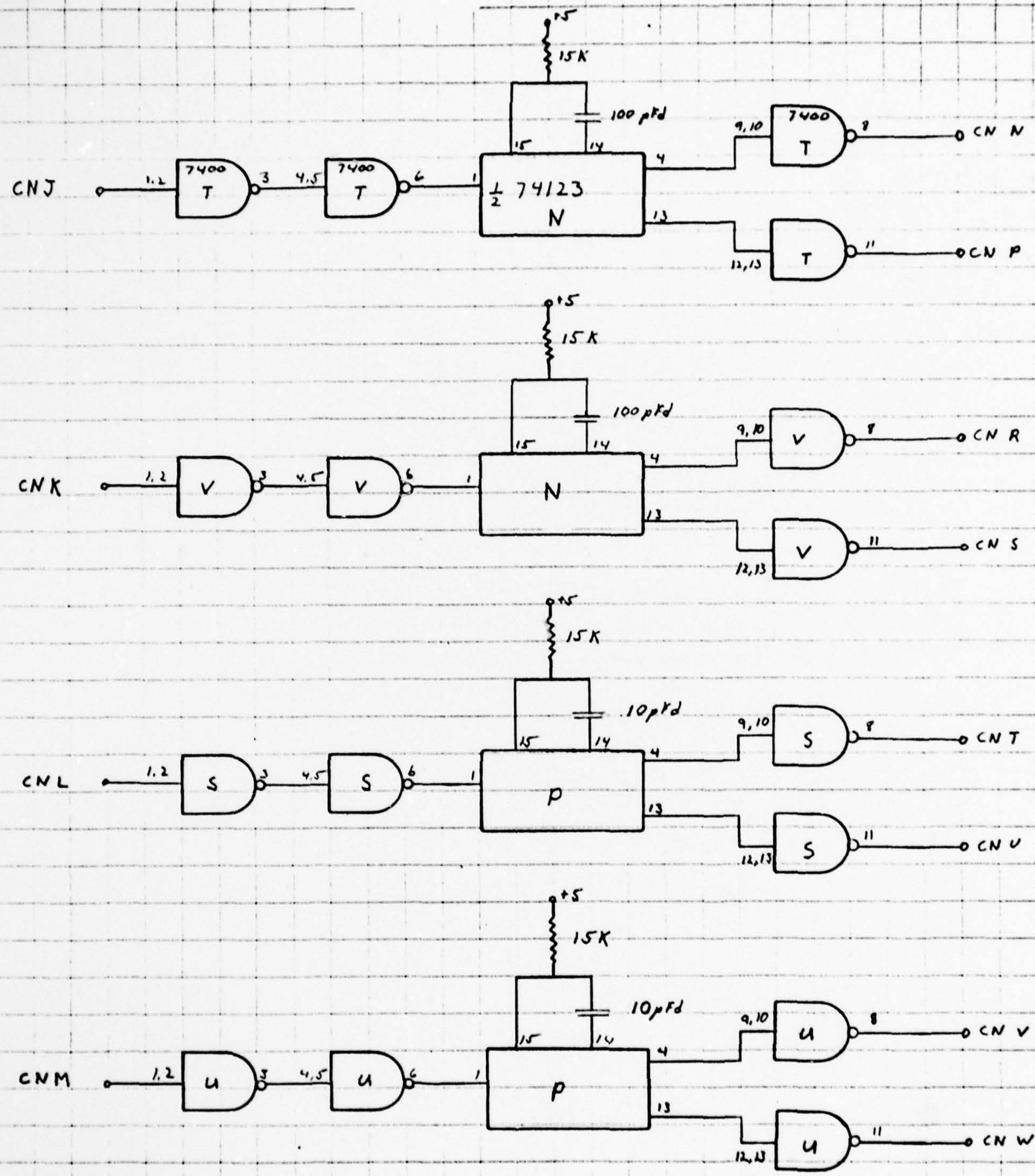
FIGURE 2.6



CARD #1

CLOCKED DELAY

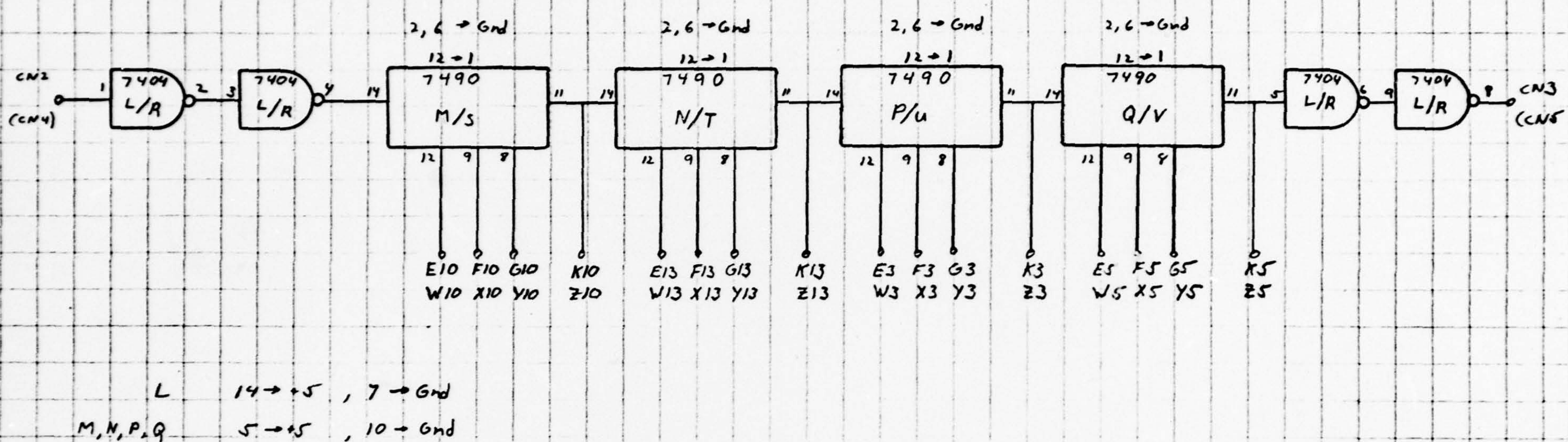
FIGURE 2.7



CARD #1

ONE SHOTS

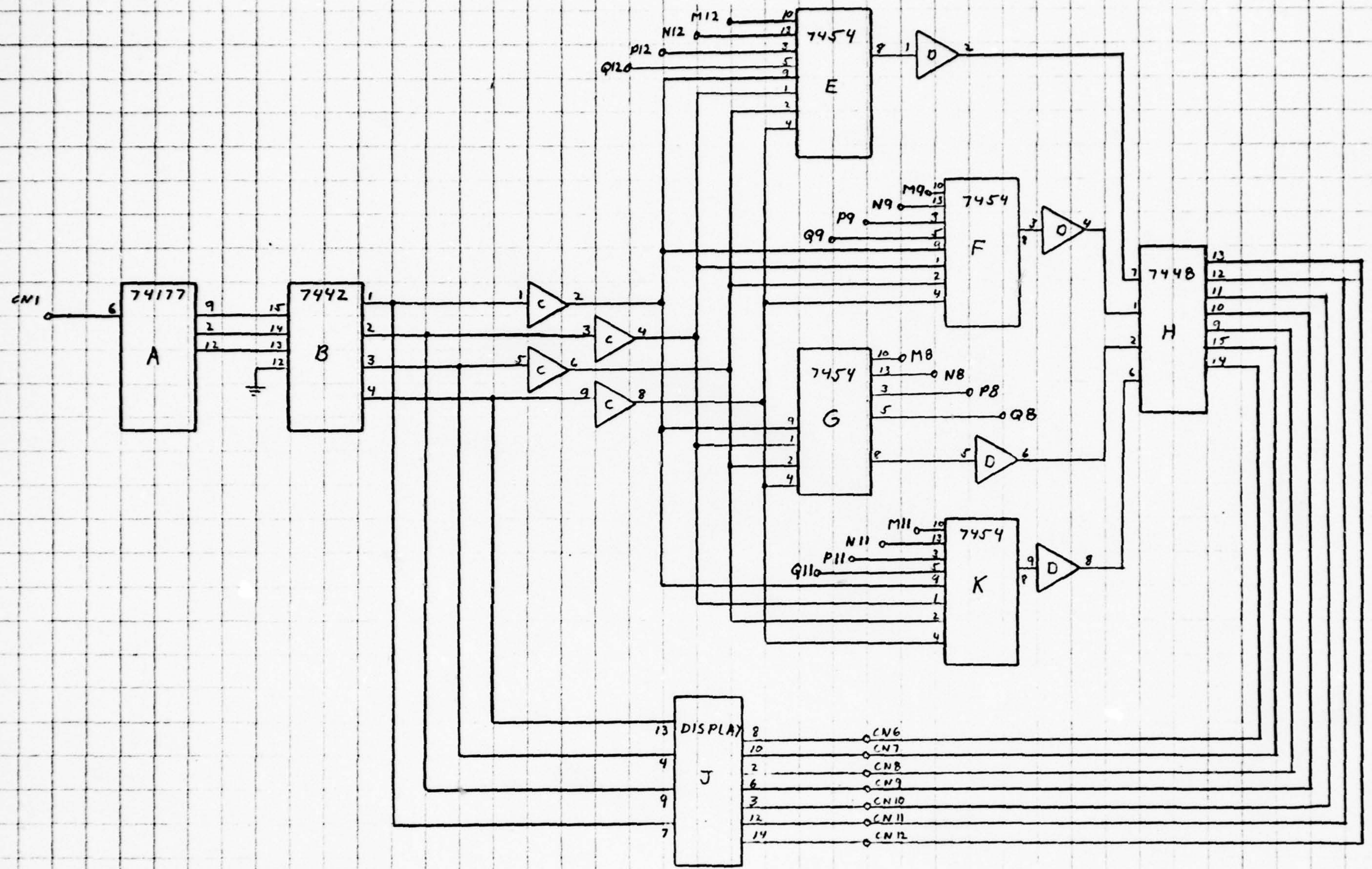
FIGURE 2.8



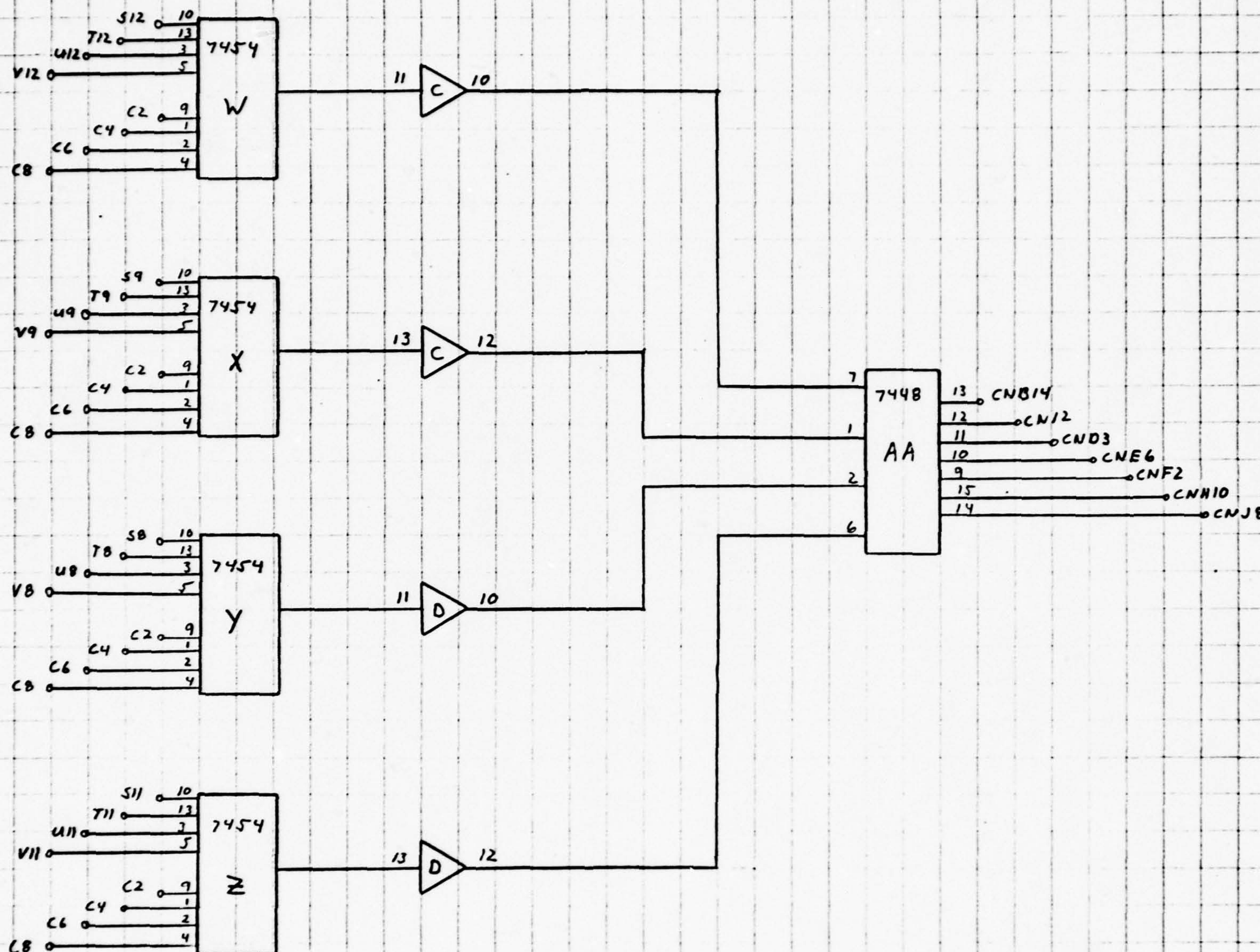
CARD #2

4-DIGIT DECIMAL COUNTER

FIGURE 2.9



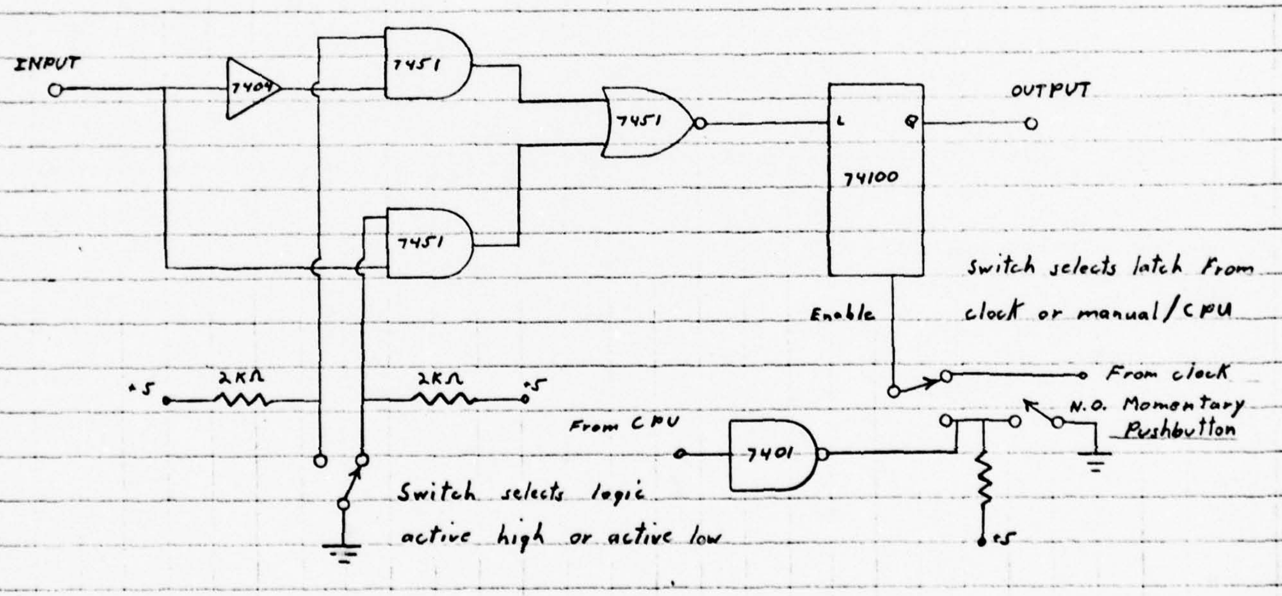
CARD #2
COUNTER DISPLAY #1
FIGURE 2.10



CARD #2

COUNTER DISPLAY #2

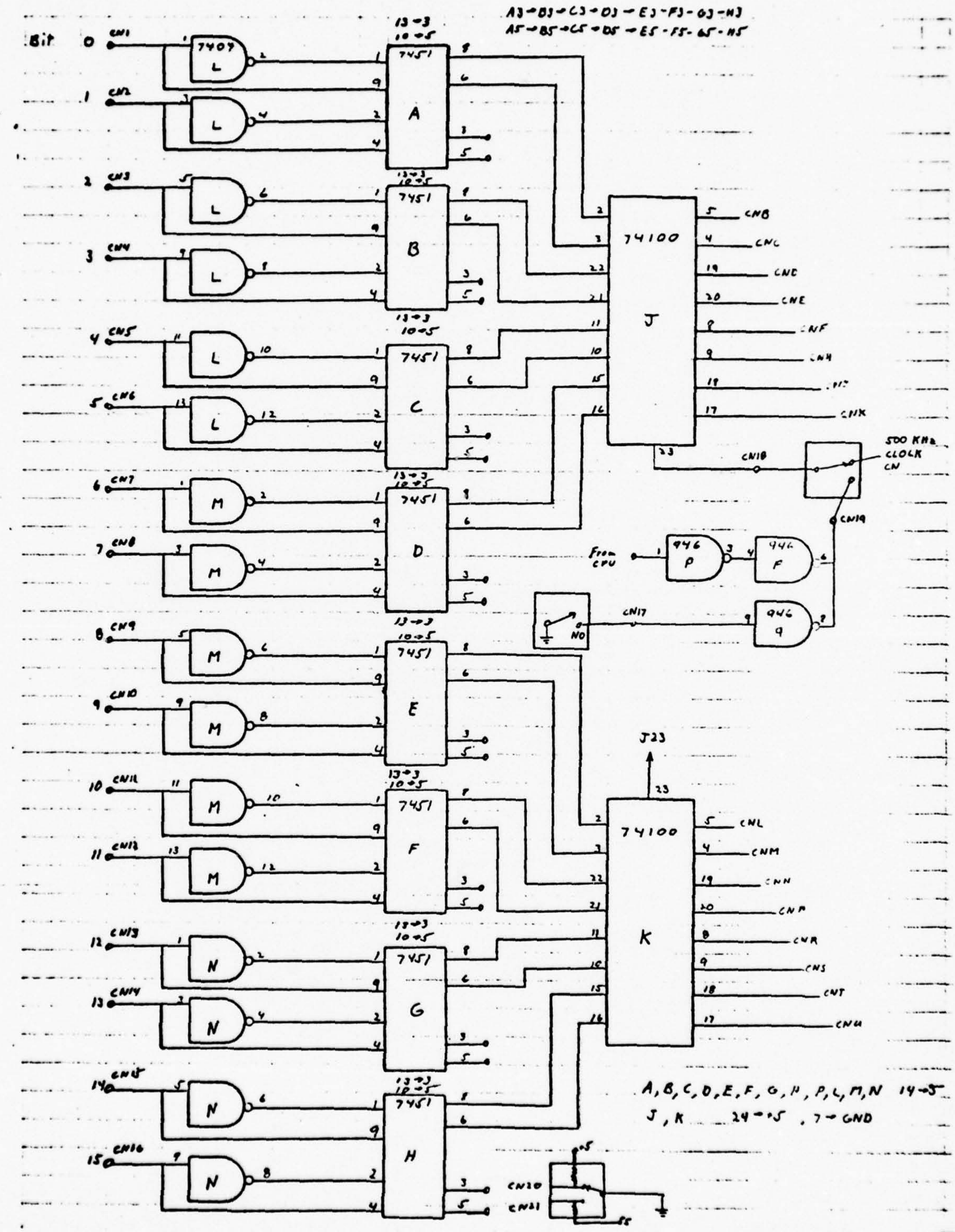
FIGURE 2.11



LATCH FOR DISCRETES IN

(Functional Block Diagram)

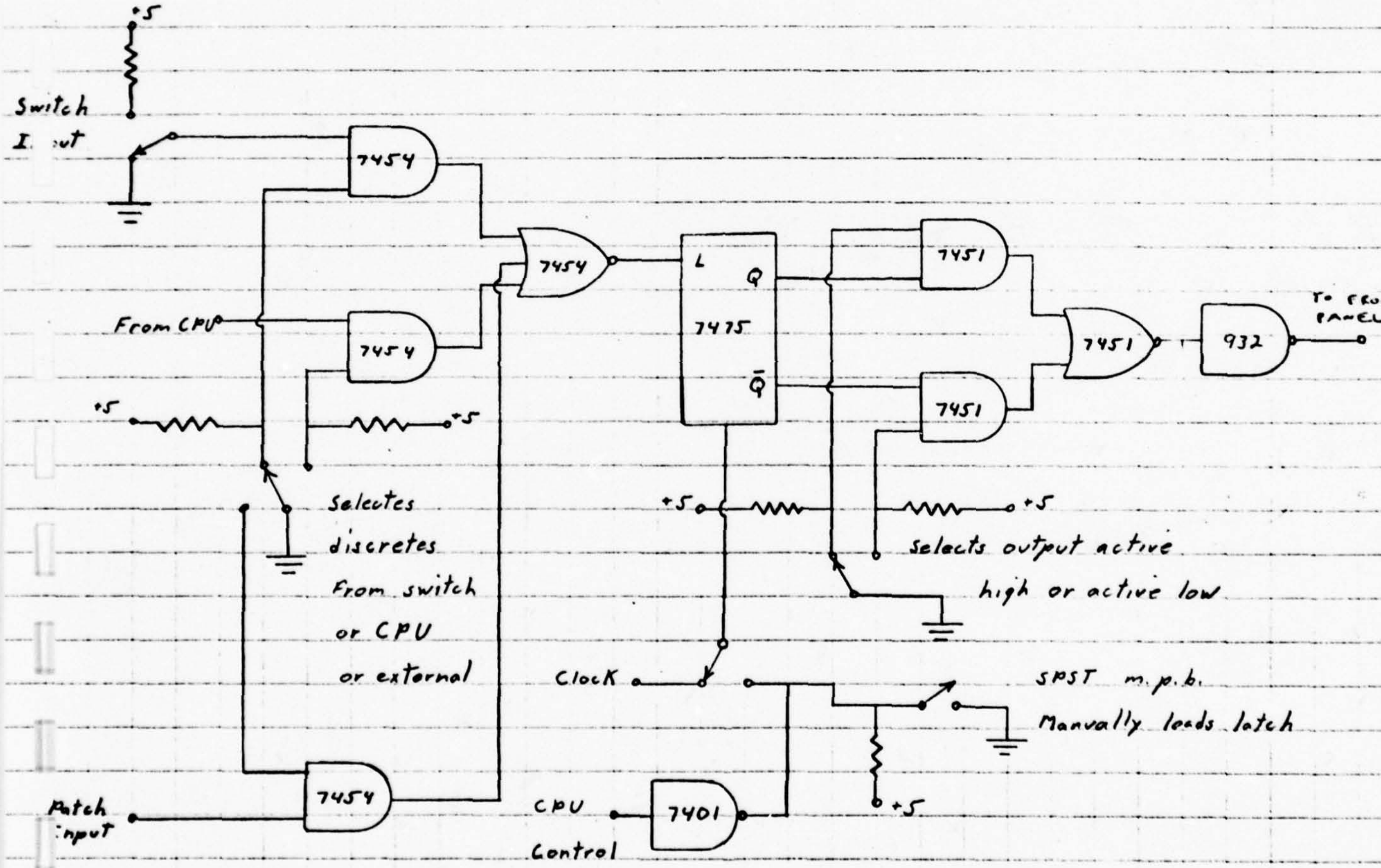
FIGURE 2.12



CARD #3

LATCH FOR DISCRETES IN

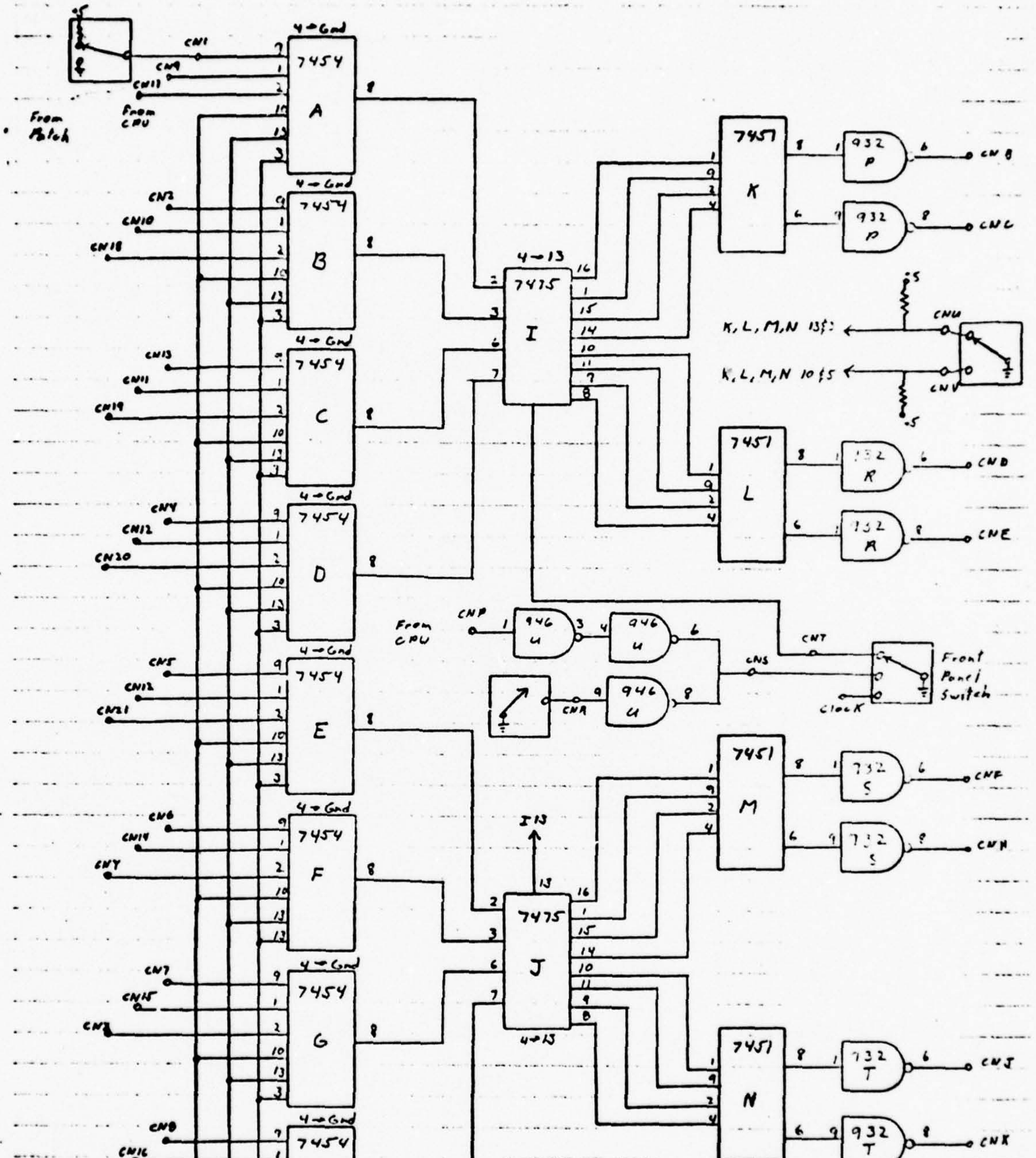
FIGURE 2.13



LATCH FOR DISCRETES OUT

(Functional Block Diagram)

FIGURE 2.14

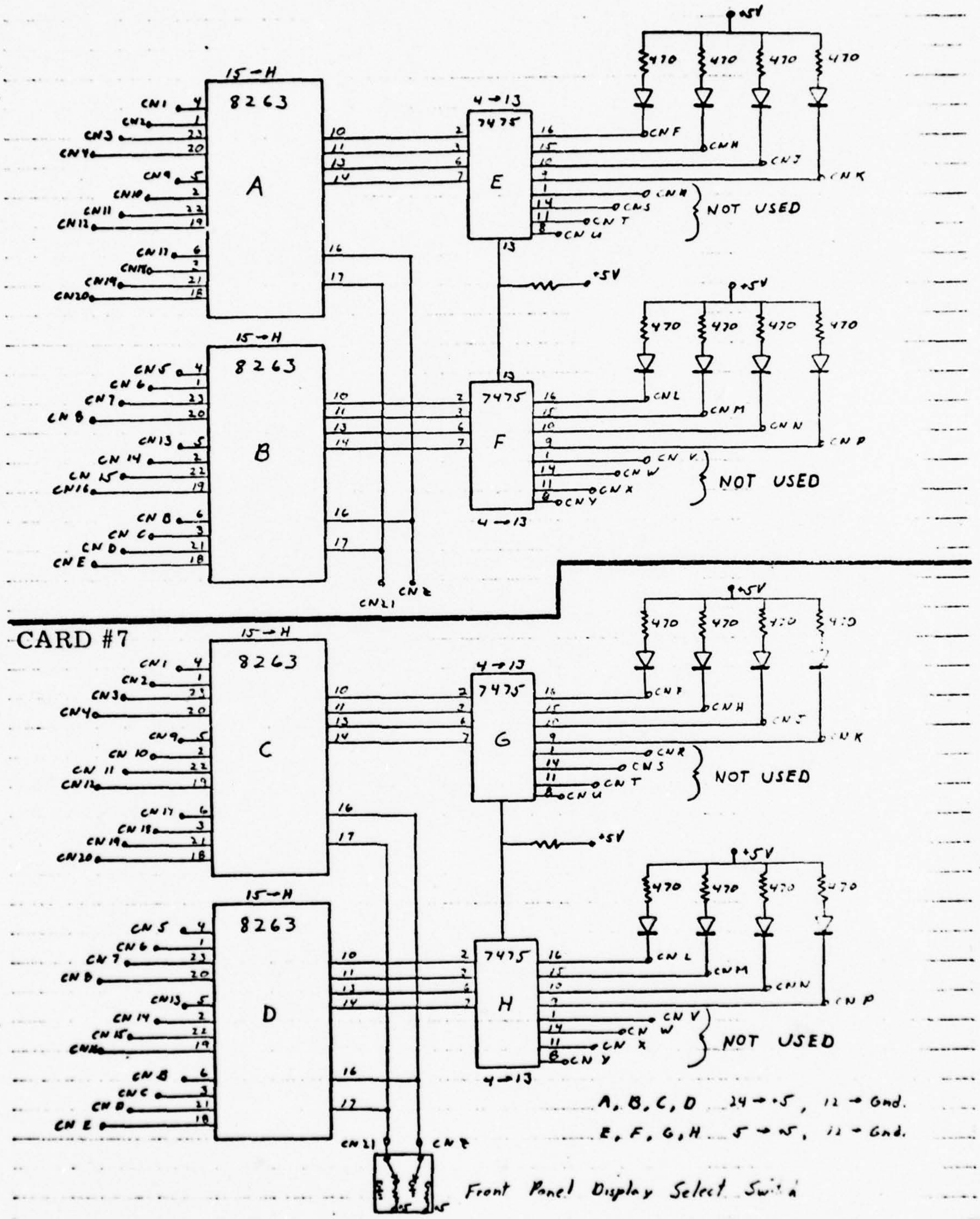


CARD #4&#5

DISCRETE OUTPUT CIRCUITRY (Two cards, each as shown above)

FIGURE 2.15

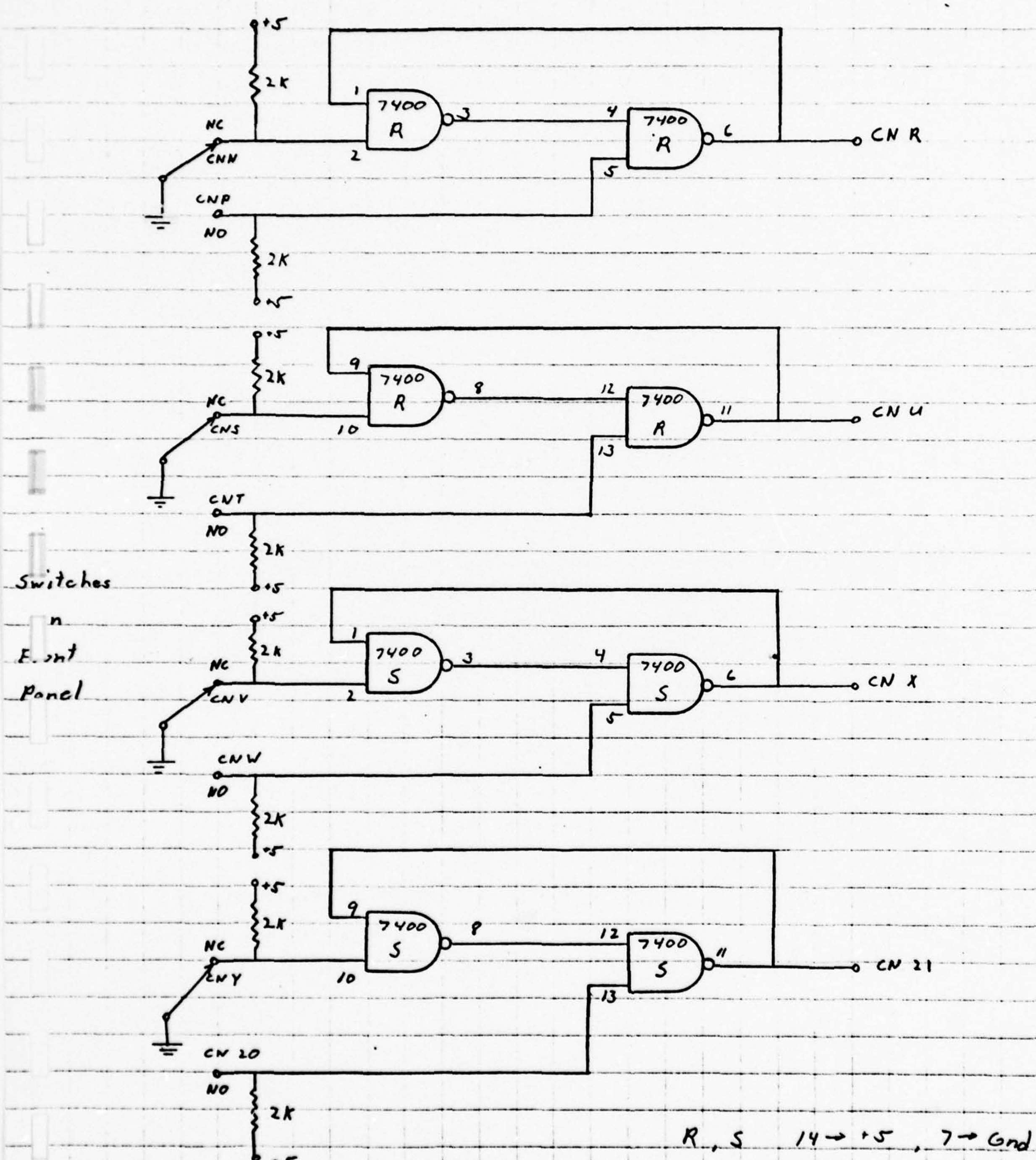
CARD #6



CARD #6&#7

DISCRETE & CPU OUTPUT DISPLAY

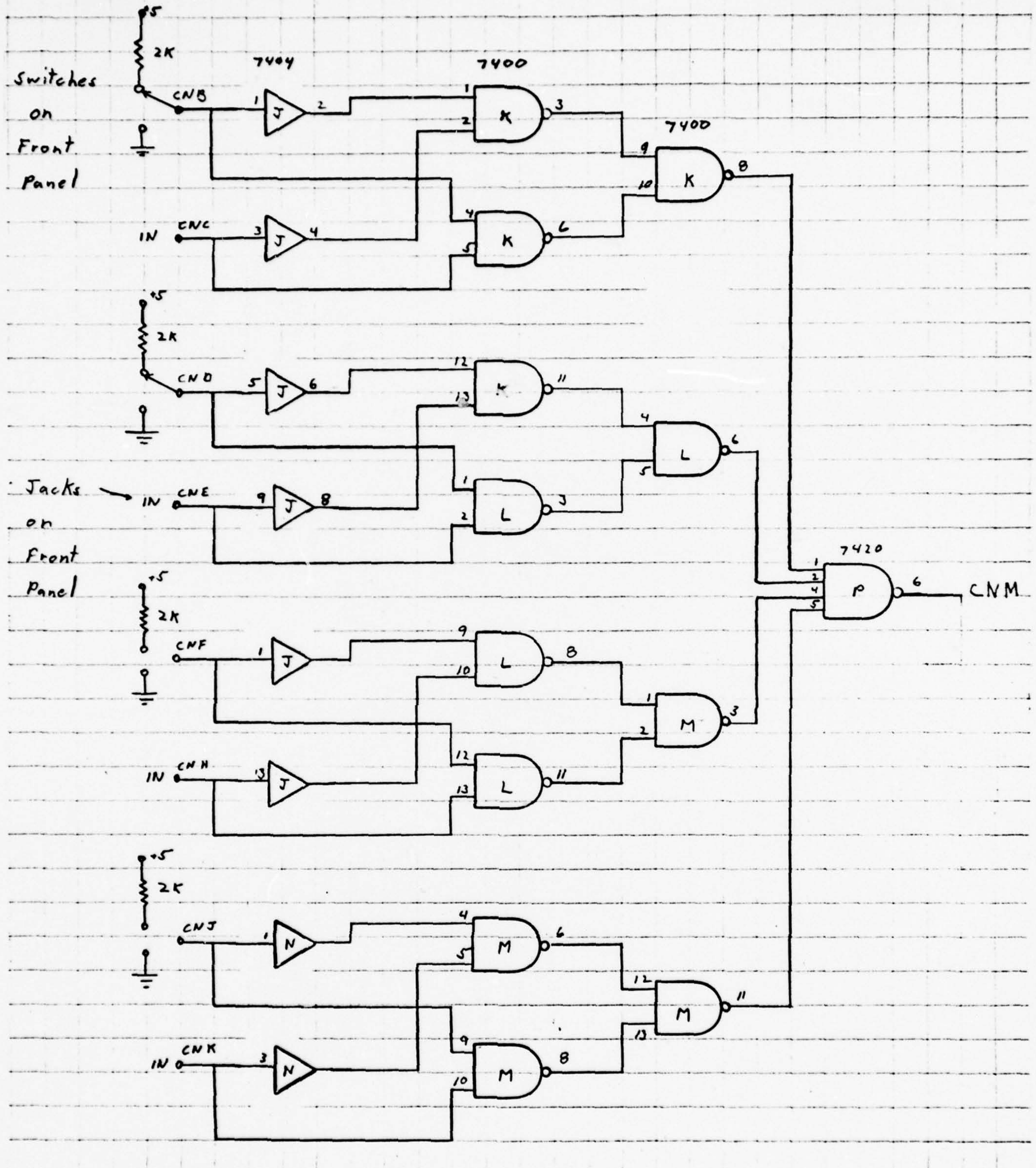
FIGURE 2.16



CARD #8

SWITCH DEBOUNCE CIRCUITS

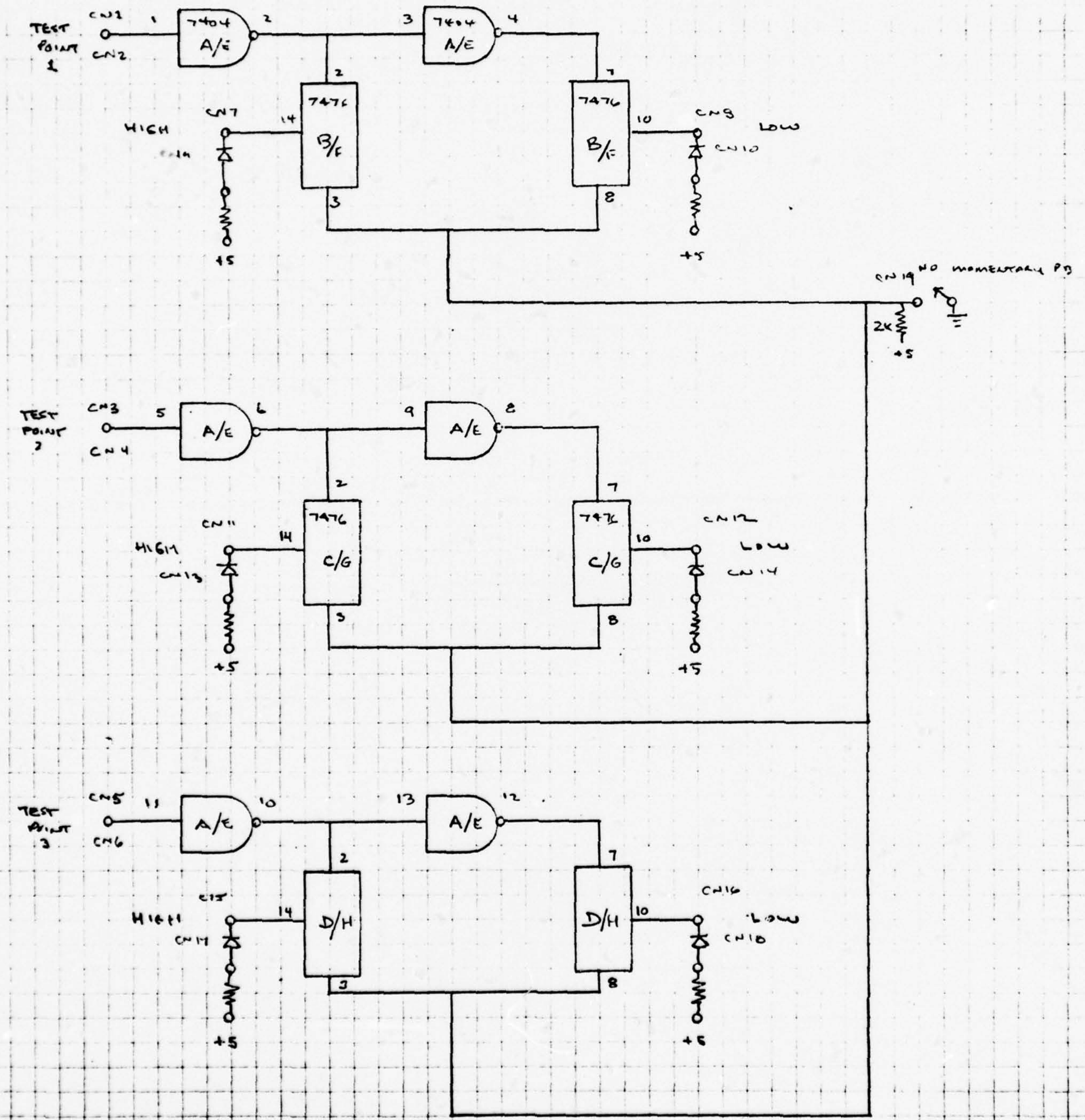
FIGURE 2.17



CARD #8

PROGRAMMABLE COINCIDENCE TRIGGER

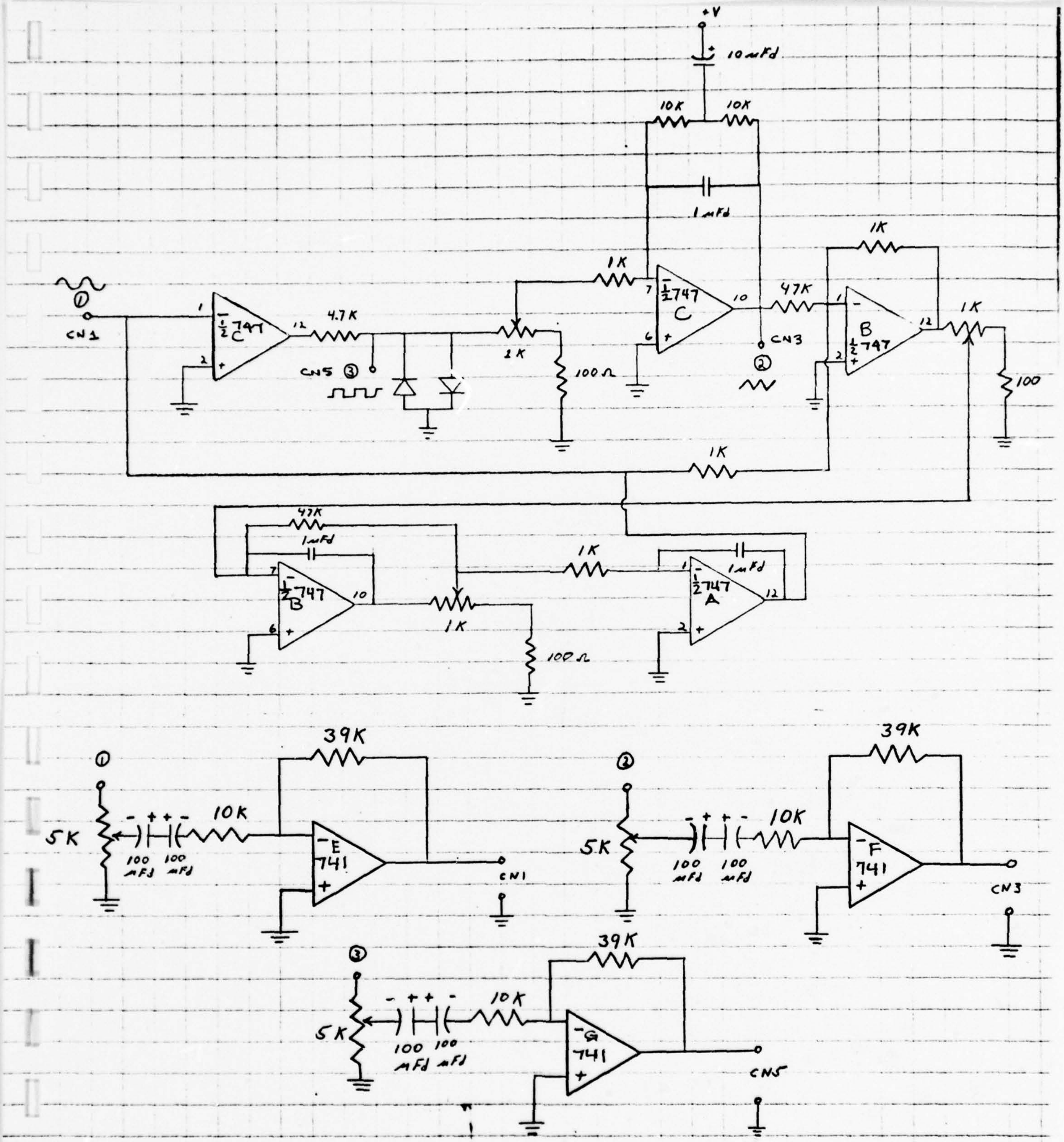
FIGURE 2.18



CARD #8

PULSE CATCHER CIRCUITRY

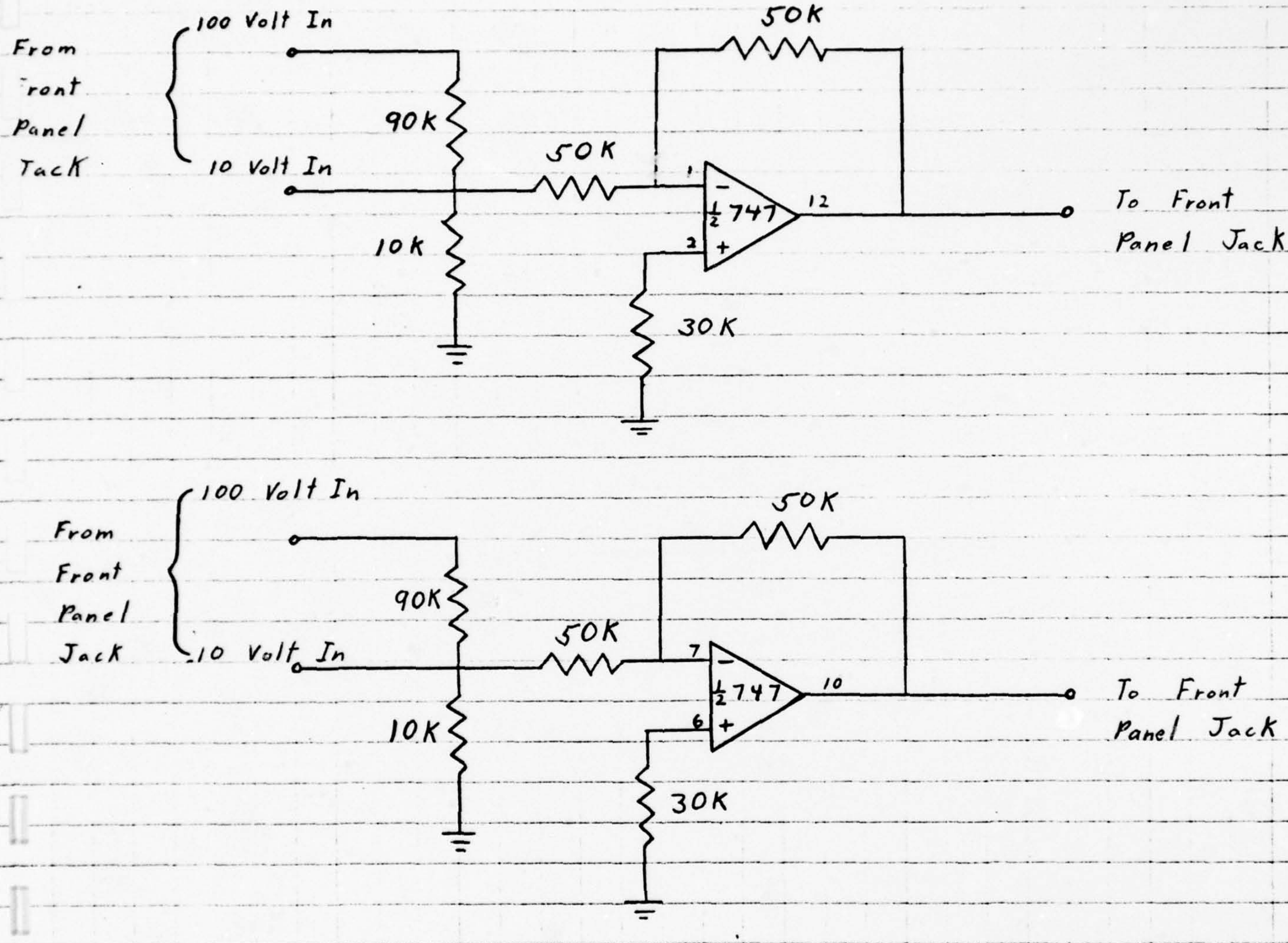
FIGURE 2.19



CARD #9

FUNCTION GENERATOR

FIGURE 2.20

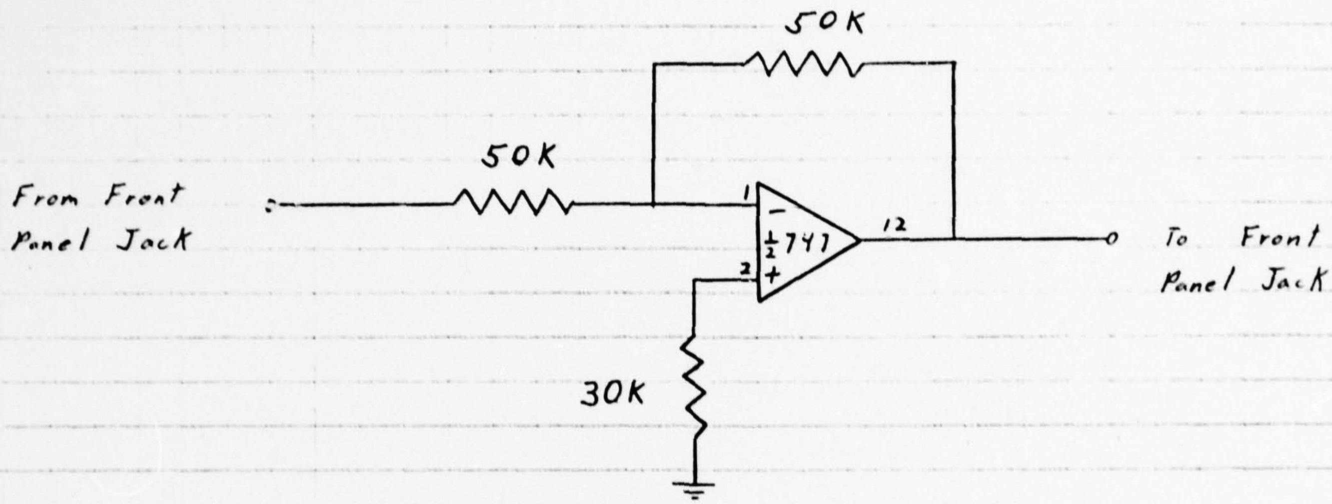


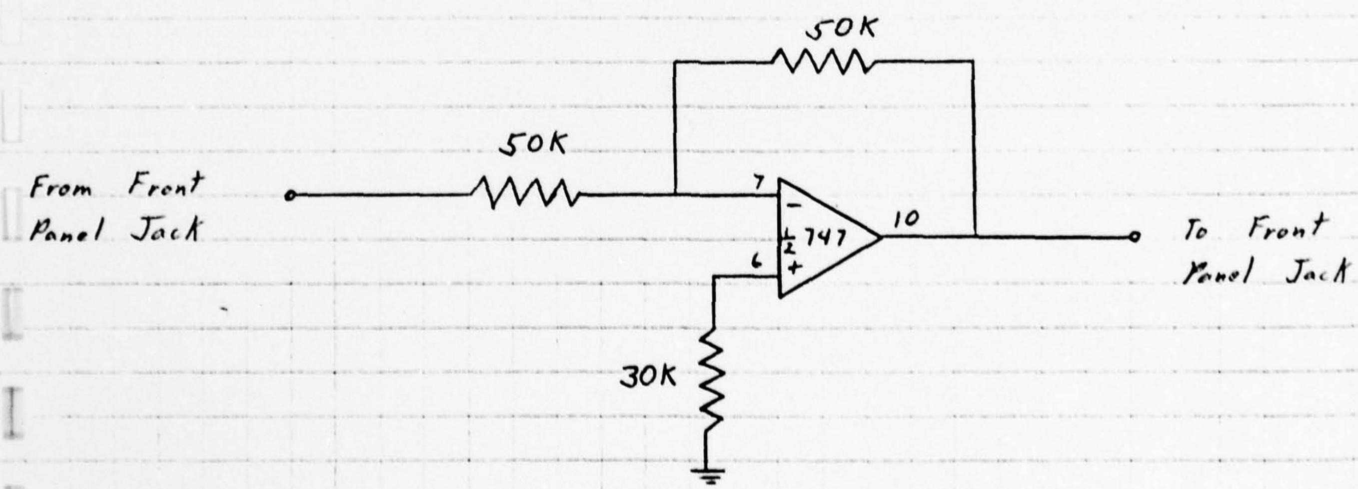
Note: 12 of the above circuits were implemented on the card.

CARD #10

RECEIVER AMPLIFIER PAIR

FIGURE 2.21





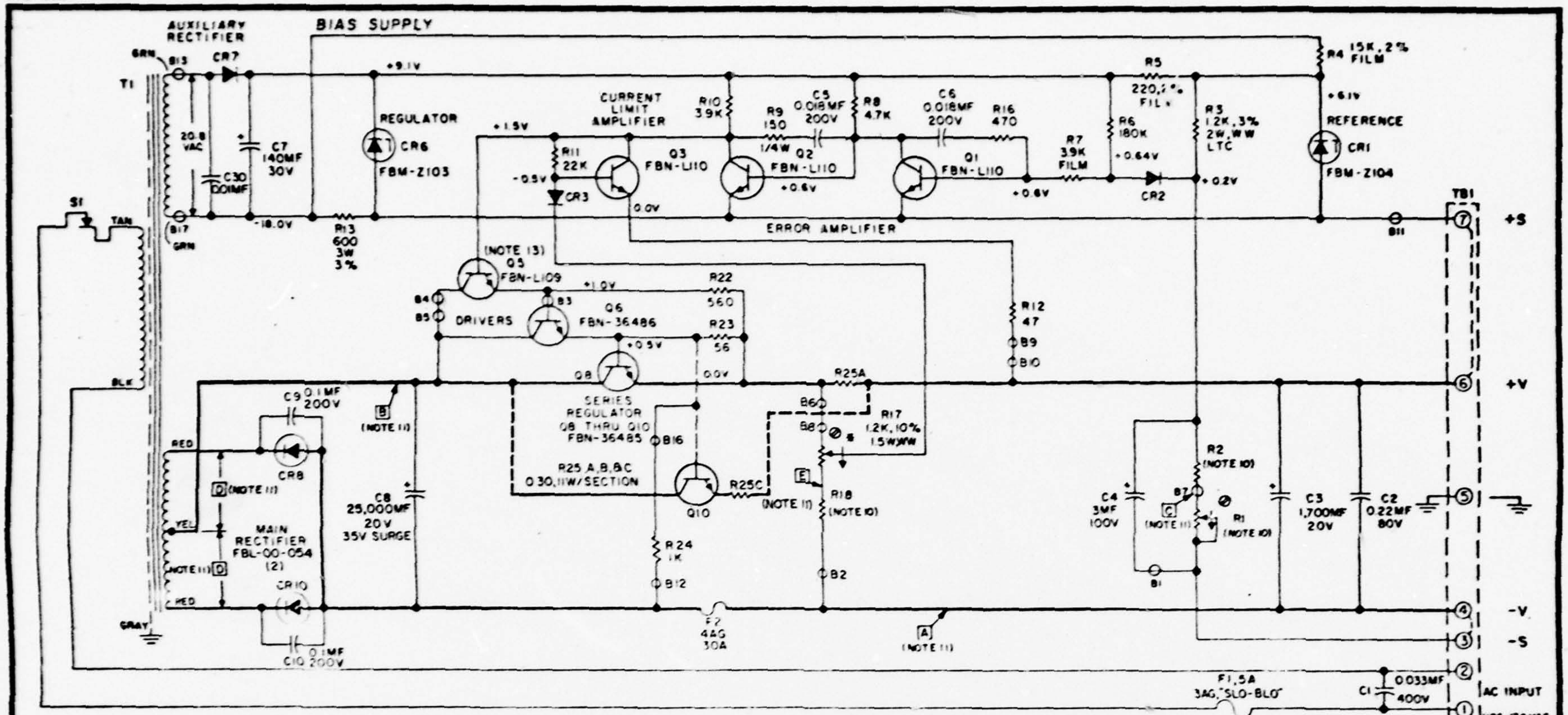
Note: 12 of the above circuits were implemented on the card.

CARD #11

TRANSMITTER AMPLIFIER PAIR

FIGURE 2.22

SCB-CC3-6A



NOTES

1. RESISTOR VALUES ARE IN OHMS.
2. RESISTOR WATTAGE 1/2 WATT; RESISTORS ABOVE 2 WATTS ARE WIREWOUND UNLESS OTHERWISE NOTED.
3. RESISTOR TOLERANCES: COMP ±10%; WIREWOUND ±5%; FILM ±5%; UNLESS OTHERWISE NOTED.
4. CAPACITOR TOLERANCES: ELECTROLYTIC -10%,+100%; MYLAR ±10%; UNLESS OTHERWISE NOTED.
5. SYMBOLS:
   - INDICATES CLOCKWISE ROTATION OF SHAFT.
   - INDICATES ADJUSTMENT OR CALIBRATION CONTROL.
   - INDICATES CONNECTION TO CHASSIS.
   - INDICATES ACTUAL UNIT MARKING
   - * SEE INSTRUCTION MANUAL.
   - LAMBDA PT. #FBL-00-030; USE IN4002 DIODE FOR REPLACEMENT UNLESS OTHERWISE NOTED.
   - INDICATES TERMINAL ON PRINTED WIRING BOARD "B"
6. DESIGNATIONS ARE LAMBDA PART NUMBERS.
7. DERATE CURRENT 10% FOR 45-55Hz OR 360-440 Hz
8. CONDITIONS FOR CIRCUIT POINT MEASUREMENTS: INPUT: 115VAC, 60 Hz, OUTPUT NOM'L VDC NO LOAD INDICATED VOLTAGES ARE TYPICAL VALUES AND ARE DC UNLESS OTHERWISE NOTED. DC MEASUREMENTS TAKEN WITH 20,000 OHMS/V VOLTMETER BETWEEN +S (TERM. 7) & INDICATED POINTS UNLESS NOTED.
9. COAT BOTH SIDES OF INSULATING WAFER WITH DOW CORNING NO. 340 SILICONE GREASE.
10. SEE TABLE I FOR COMPONENT VALUES.
11. SEE TABLE I FOR VOLTAGE VALUES.
12. IF Q5 IS REPLACED, ALWAYS RETAIN RADIATOR AND INSTALL WITH NEW TRANSISTOR.

REFERENCE DESIGNATIONS NOT USED	
Q4	C11
Q7	C12
	R14
	R15
	R19-21
CR4,5	CR9

MODELS
LM-CC3
LM-CC3P3
LM-CC3P6
LM-CC4P5
LM-CC5
LM-CC6
LM-CC4

FOR WIRING OF POWER SUPPLY TO LOAD REFER TO SUPPLY-TO-LOAD WIRING DIAGRAMS.

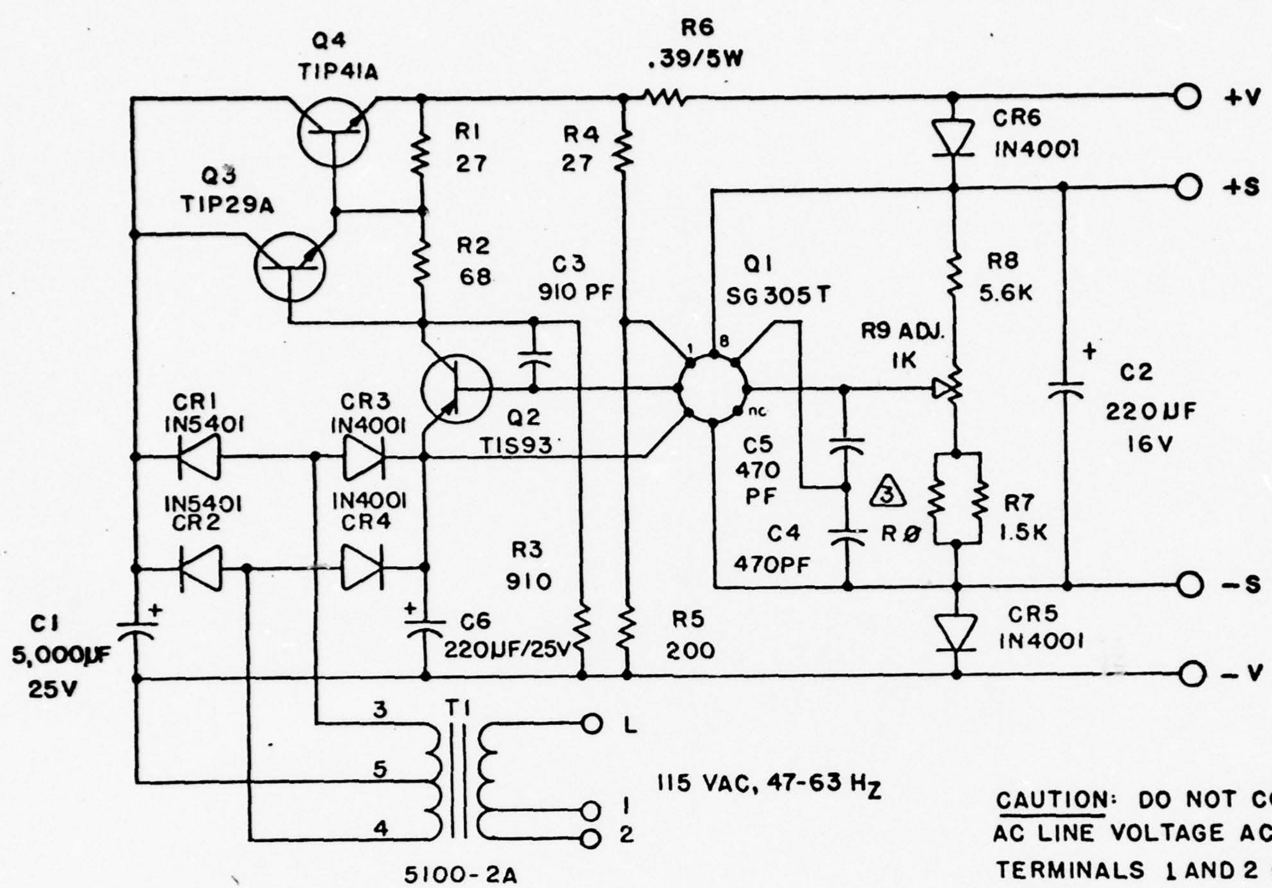
DOTTED CONNECTIONS SHOWN ON TB1 INDICATE JUMPERS IN PLACE FOR LOCAL SENSING CONNECTION.

THIS SCHEMATIC APPLIES TO UNITS BEARING SERIAL NO. PREFIXES A,B

SCHEMATIC DIAGRAM
REGULATED POWER SUPPLY

LAMBDA ELECTRONICS CORP.
MELVILLE NEW YORK

FIGURE 2.23
Five Volt, 8 Amp Power Supply



CAUTION: DO NOT CONNECT AC LINE VOLTAGE ACROSS TERMINALS 1 AND 2 OF TRANSFORMER (T1). DAMAGE WILL RESULT.

NOTES: UNLESS OTHERWISE SPECIFIED

1. RESISTORS SPECIFIED IN OHMS
2. FIXED RESISTORS ARE 1/2 WATT, 5%
3. (△) FACTORY SELECTED
4. TYPE AND VALUES SUBJECT TO CHANGE

TITLE: MODULAR DC POWER SUPPLY MODEL: 5 RS-23	DWG NO: 5100-102
	REV:
WORTEK, INC. 5971 RESEDA BOULEVARD TARZANA, CALIFORNIA	DWN BY:
	APPR:
	REL DATE 1-30-74

FIGURE 2.24 Five Volt, Three Amp Power Supply

*NOTE-This supply is also used to provide ± 14 volt at 150 ma for analog op amps

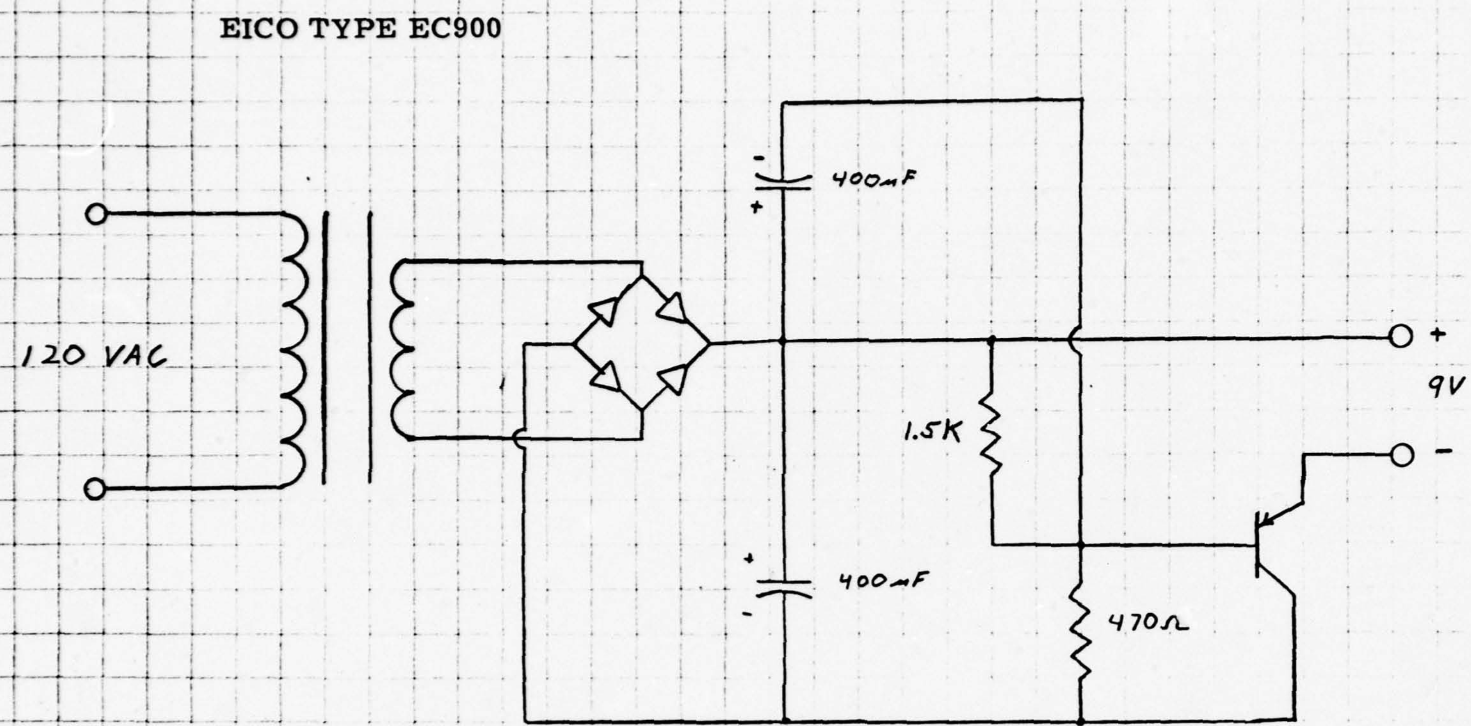


FIGURE 2.25

Nine Volt*, 250 ma Power Supply

NUMBER	FUNCTION
Card #1	Clock, Interrupt, One Shots and Clocked Delays
Card #2	Counters and Counter Display
Card #3	Discretes IN
Card #4	Discretes OUT
Card #5	Discretes OUT
Card #6	Display
Card #7	Display
Card #8	Programmable Trigger, Switch Debouncer and Pulse Catcher
Card #9	Analog Function Generator
Card #10	Analog Receiver
Card #11	Analog Driver

TABLE 2.1

CARD FUNCTION LIST

CARD #1

FUNCTION	SIGNAL	CARD PIN NUMBER	TO	CONNECTOR #36
Clock and	500KHz	15		1
Interrupt	250KHz	16		2
Generator	100KHz	17		3
	50KHz	18		4
	10KHz	19		5
	1KHz	20		9
	100Hz	21		8
	10Hz	13		6
	1Hz	14		7
				CONNECTOR #37
Delay #1	IN	1		1
	CLOCK	2		2
	OUT	3		3
Delay #2	IN	4		4
	CLOCK	5		5
	OUT	6		6
Delay #3	IN	7		7
	CLOCK	8		9
	OUT	9		8
				CONNECTOR #38
Delay #4	IN	10		1
	CLOCK	11		2
	OUT	12		3
Delay #5	IN	B		4
	CLOCK	C		5
	OUT	D		6
Delay #6	IN	E		7
	CLOCK	F		9
	OUT	H		8
				CONNECTOR #60
One Shot #1	IN 1	J		1
		N		2
		P		3
One Shot #2	IN 2	K		4
		R		5
		S		6
One Shot #3	IN 3	L		7
		T		8
		U		9
One Shot #4	IN 4	M		10
		V		11
		W		12

TABLE 2.2
CLOCK, DELAYS AND ONE-SHOTS

FUNCTION	SIGNAL	CARD PIN NUMBER	TO	CONNECTOR #58
Counter #1	IN	2		4
	OUT	3		5
	CLEAR	17		6
Counter #2	IN	4		1
	OUT	5		2
	CLEAR	18		3
				9 PIN CONNECTOR
Display #1	segment a	12		1
	b	11		2
	c	10		3
	d	9		4
	e	8		5
	f	7		6
	g	6		7
	cathode	13		8
	cathode	14		9
				9 PIN CONNECTOR
Display #2	segment a	B		1
	b	C		2
	c	D		3
	d	E		4
	e	F		5
	f	H		6
	g	J		7
	cathode	15		8
	cathode	16		9

TABLE 2.3

COUNTERS AND COUNTER DISPLAY

CARD #2

CARD PIN NUMBER			CONNECTOR #59
1	LSB		1
2			2
3			3
4			4
5			5
6			6
7			7
8		INPUTS	9
9		(from front panel)	10
10			11
11			12
12			13
13			14
14			15
15			16
16	MSB		17
B	LSB		18
C			19
D			20
E			21
F			22
H			23
J			24
K		OUTPUTS	26
L		(to front panel)	27
M			28
N			29
P			30
R			31
S			32
T			33
U	MSB		34

TABLE 2.4

DISCRETES IN

CARD #3

CARD #3 (continued)

CARD CONNECTOR		CONNECTOR #55	
18	Latch Control	3	from switch pole for latch select
19		4	to front panel switch (select manual latch or CPU)
17		5	to front panel switch manual latch NO/SP/ MPB from CPU (active high)
20	Logic Select	1	to front panel switch (high in gives high out)
21		2	to front panel switch low in gives high out
22		6	+5 volts from card #3

TABLE 2.4 CONT.

CARD CONNECTOR	FUNCTION		TO	CONNECTOR		
				CONN#70		CONN#71
1		(LSB)		1		1
2		(low order bits on card 4		2		2
3		high order bits on card 5)		3		3
4	INPUT 1			4		4
5	(switches)			5		5
6				6		6
7				7		7
8		(MSB)		8		8
9		(LSB)		9		9
10				10		10
11				11		11
12	INPUT 2			12		12
13	(patch holes)			13		13
14				14		14
15				15		15
16		(MSB)		16		16
17		(LSB)	CONN#63	1	CONN#65	1
18				2		2
19				3		3
20	INPUT 3			4		4
21	(computer)			5		5
Y				6		6
X				7		7
W		(MSB)		8		8
B		(LSB)	CONN#48	1	CONN#49	1
C				2		2
D				3		3
E	OUTPUTS			4		4
F				5		5
H				6		6
J				7		7
K		(MSB)		8		8
T	to switch	Latch strobe select line			36	
S	to switch	CPU & manual strobe			35	
				CONNECTOR #43		
R		Manual strobe switch, active low			3	
P		From CPU, active low				
U		NG logic select				
V		NO logic select				
				CONNECTOR #46		
L		Select Input 1 active high				
M		Select Input 2 active high				
N		Select Input 3 active high				

TABLE 2.5
DISCRETES OUT
CARD #4&#5

	Card Connector		Card Connector	
A Input (Discretes Out)				
	1		B	
	2		C	
	3	Low	D	
Card 6	4	Order	E	Card 4
	5	Bits	F	
	6		H	
	7		J	
	8	(back plane wiring)	K	
	1		B	
	2		C	
	3	High	D	
Card 7	4	Order	E	Card 5
	5	Bits	F	
	6		H	
	7		J	
	8		K	
B Input (Discretes In)				
	9		B	
	10		C	
	11	Low	D	
Card 6	12	Order	E	Card 3
	13	Bits	F	
	14		H	
	15		J	
	16	(back plane wiring)	K	
	9		L	
	10		M	
	11	High	N	
Card 7	12	Order	P	Card 3
	13	Bits	R	
	14		S	
	15		T	
	16		U	

TABLE 2.6
DISCRETE DISPLAY
CARDS #6&#7

	CARD CONNECTOR		CARD CONNECTOR	
C Input (computer)				
	17		1	
	18	Low	2	
	19	Order Bits	3	
	20		4	
Card 6	B		5	Connector #66
	C		6	
	D		7	
	E		8	
	17		1	
	18		2	
	19		3	
	20	High	4	
Card 7	B	Order Bits	5	Connector #64
	C		6	
	D		7	
	E		8	
Outputs to LED's				
Card 6	F		1	8 pin connector
	H		2	
	J		3	
	K		4	
	L		5	
	M		6	
	N		7	
	P		8	
Card 7	F		1	8 pin connector
	H		2	
	J		3	
	K		4	
	L		5	
	M		6	
	N		7	
	P		8	
Display Select Bits				
Card 6	21	(back plane wiring)	21	Card 7
	Z		Z	
Card 7	21		1	Connector #43
	Z		2	

TABLE 2.6 CONT.
CARDS #6&#7

FUNCTION	SIGNAL	CARD PIN NUMBER	TO	CONNECTOR #42
Pulse Catchers (6)	Input 1	1		1
	Input 2	2		3
	Input 3	3		5
	Input 4	4		7
	Input 5	5		9
	Input 6	6		11
	Clear	19		12
				CONNECTOR #47
Light Outputs	1 High	7		1
	1 Low	8		2
	2 High	9		3
	2 Low	10		4
	3 High	11		5
	3 Low	12		6
	4 High	13		7
	4 Low	14		8
	5 High	15		9
	5 Low	16		10
	6 High	17		11
	6 Low	18		12
				CONNECTOR #39
Programmable Trigger				
	Switch 1	B		1
	In 1	C		2
	Switch 2	D		3
	In 2	E		4
	Switch 3	F		5
	In 3	H		6
	Switch 4	J		7
	In 4	K		8
	Out	L		9

TABLE 2.7
CARD #8
PULSE CATCHERS, TRIGGER AND SWITCHES

FUNCTION	SIGNAL	CARD PIN NUMBER	TO	CONNECTOR #44
Switch Debounce #1	NC	N		1
	NO	P		2
	OUT	R		3
#2	NC	S		4
	NO	T		5
	OUT	U		6
#3	NC	V		7
	NO	W		8
	OUT	X		9
#4	NC	Y		10
	NO	20		11
	OUT	21		12

TABLE 2.7 CONT.

CARD #8

CARD CONNECTOR	FUNCTION	CONNECTOR #59
CN 1, CN R	Sine	1
CN 3, CN D	Square	3
CN 5, CN F	Triangle	4

TABLE 2.8

FUNCTION GENERATOR
CARD #9

		Card Conn.	Conn. #57 (to front panel)
Chip A	100 V in	CN 1	1
	10 V in	2	2
	Output (pin 12)	3	3
	100 V in	4	4
	10 V in	5	5
	Output (pin 10)	6	6
Chip B	100 V in	7	7
	10 V in	8	8
	Output (pin 12)	9	9
	100 V in	X	10
	10 V in	Y	11
	Output (pin 10)	Z	12
Chip C	100 V in	13	13
	10 V in	14	14
	Output (pin 12)	15	15
	100 V in	16	16
	10 V in	17	17
	Output (pin 10)	18	18
Chip D	100 V in	19	19
	10 V in	20	20
	Output (pin 12)	21	21
	100 V in	B	22
	10 V in	C	23
	Output (pin 10)	D	24
Chip E	100 V in	E	25
	10 V in	F	26
	Output (pin 12)	H	27
	100 V in	J	28
	10 V in	K	29
	Output (pin 10)	L	30

TABLE 2.9
ANALOG RECEIVER
CARD #10

Chip F	100 V in	M	31
	10 V in	N	32
	Output (pin 12)	P	33
	100 V in	R	34
	10 V in	S	35
	Output (pin 10)	T	36
	-15 Volts	10	
	GROUND	11	
	+15 Volts	12	

TABLE 2.9 CONT.
CARD #10

		Card Connector	Connector #59 (to front panel)
Chip A	IN	CN M	22
	OUT	CN 14	24
	IN	CN 15	25
	OUT	CN 16	26
Chip B	IN	CN 5	11
	OUT	CN 6	12
	IN	CN 7	15
	OUT	CN 8	16
Chip C	IN	CN B	31
	OUT	CN C	32
	IN	CN D	33
	OUT	CN E	34
Chip D	IN	CN 17	27
	OUT	CN 18	28
	IN	CN 19	29
	OUT	CN 20	30
Chip E	IN	CN F	17
	OUT	CN H	18
	IN	CN J	19
	OUT	CN K	20
Chip F	IN	CN 1	5
	OUT	CN 2	7
	IN	CN 3	9
	OUT	CN 4	10

TABLE 2.10

ANALOG TRANSMITTER
CARD # 11

2.3.1.1.3 ADDRESS STACK

The address stack contains eight 14-bit registers providing storage for eight lower and six higher order address bits in each register. One register is used as the program counter (storing the effective address) and the other seven permit address storage for nesting of subroutines up to seven levels. The stack automatically stores the content of the program counter upon the execution of a CALL instruction and automatically restores the program counter upon the execution of a RETURN. The CALLs may be nested and the registers of the stack are used as last in/first out pushdown stack. A three-bit address pointer is used to designate the present location of the program counter. When the capacity of the stack is exceeded the address pointer recycles and the content of the lowest level register is destroyed. The program counter is incremented immediately after the lower order address bits are sent out. The higher order address bits are sent out at T2 and then incremented if a carry resulted from T1. The 14-bit program counter provides direct addressing of 16K bytes of memory. Through the use of an I/O instruction for bank switching, memory may be indefinitely expanded.

2.3.1.1.4 SCRATCH PAD MEMORY OR INDEX REGISTERS

The scratch pad contains the accumulator (A register) and six additional 8-bit registers (B, C, D, E, H, L). All arithmetic operations use the accumulator as one of the operands. All registers are independent and may be used for temporary storage. In the case of instructions which require operations with a register in external memory, scratch pad registers H&L provide indirect addressing capability; register L contains the eight lower order bits of address and register H contains the six higher order bits of address (in this case bit 6 and bit 7 are "don't cares").

2.3.1.1.5 ARITHMETIC/LOGIC UNIT (ALU)

All arithmetic and logical operations (ADD, ADD with carry, SUBTRACT, SUBTRACT with borrow, AND, EXCLUSIVE OR, OR, COMPARE, INCREMENT, DECREMENT) are carried out in the 8-bit parallel arithmetic unit which includes carry-look-ahead logic. Two temporary registers, register "a" and register "b", are used to store the accumulator and operand for ALU operations. In addition, they are used for temporary address and data storage during intra-processor transfers. Four control bits, carry flip-flop (c), zero flip-flop (z), sign flip-flop (s), and parity flip-flop (p), are set as the result of each arithmetic and logical operation. These bits provide conditional branching capability through CALL, JUMP, or RETURN on condition instructions. In addition, the carry bit provides the ability to do multiple precision binary arithmetic.

2.3.1.1.6 I/O BUFFER

This buffer is the only link between the processor and the rest of the system. Each of the eight buffers is bi-directional and is under control of the instruction register and state timing. Each of the buffers is low power TTL compatible on the output and TTL compatible on the input.

2.3.1.2 PROCESSOR TIMING

The 8008 is a complete central processing unit. The internal organization is centered around an 8-bit internal data bus. All communication within the processor and with external components occurs on this bus in the form of 8-bit bytes of address, instruction or data. (Refer to the accompanying block diagram for the relationship of all the internal elements of the processor to each other and to the data bus.) For the 8008 a logic "1" is defined as a high and a logic "0" is defined as a low level.

2.3.1.2.1 STATE CONTROL CODING

The processor controls the use of the data bus and determines whether it will be sending or receiving data. State signals S_0, S_1, and S_2, along with SYNC inform the peripheral circuitry of the state of the processor. A table of the binary state codes and the designated state names is shown in Table 2.11.

2.3.1.2.2 TIMING

Typically, a machine cycle consists of five states, two states in which an address is sent to memory (T1 and T2), one for the instruction or data fetch (T3), and two states for the execution of the instruction (T4 and T5). If the processor is used with slow memories, the READY line synchronizes the processor with the memories. When the memories are not available for either sending or receiving data, the processor goes into the WAIT state. Figure 2.27 illustrates the processor activity during a single cycle.

The receipt of an INTERRUPT is acknowledged by the T11. When the processor has been interrupted, this state replaces T1. A READY is acknowledged by T3. The STOPPED state acknowledges the receipt of a HALT instruction.

Many of the instructions for the 8008 are multi-cycle and do not require the two execution states, T4 and T5. As a result, these states are omitted when they are not needed and the 8008 operates asynchronously with respect to the cycle length.

2.3.1.2.3 CYCLE CONTROL CODING

As previously noted, instructions for the 8008 require one, two, or three machine cycles for complete execution. The first cycle is always an

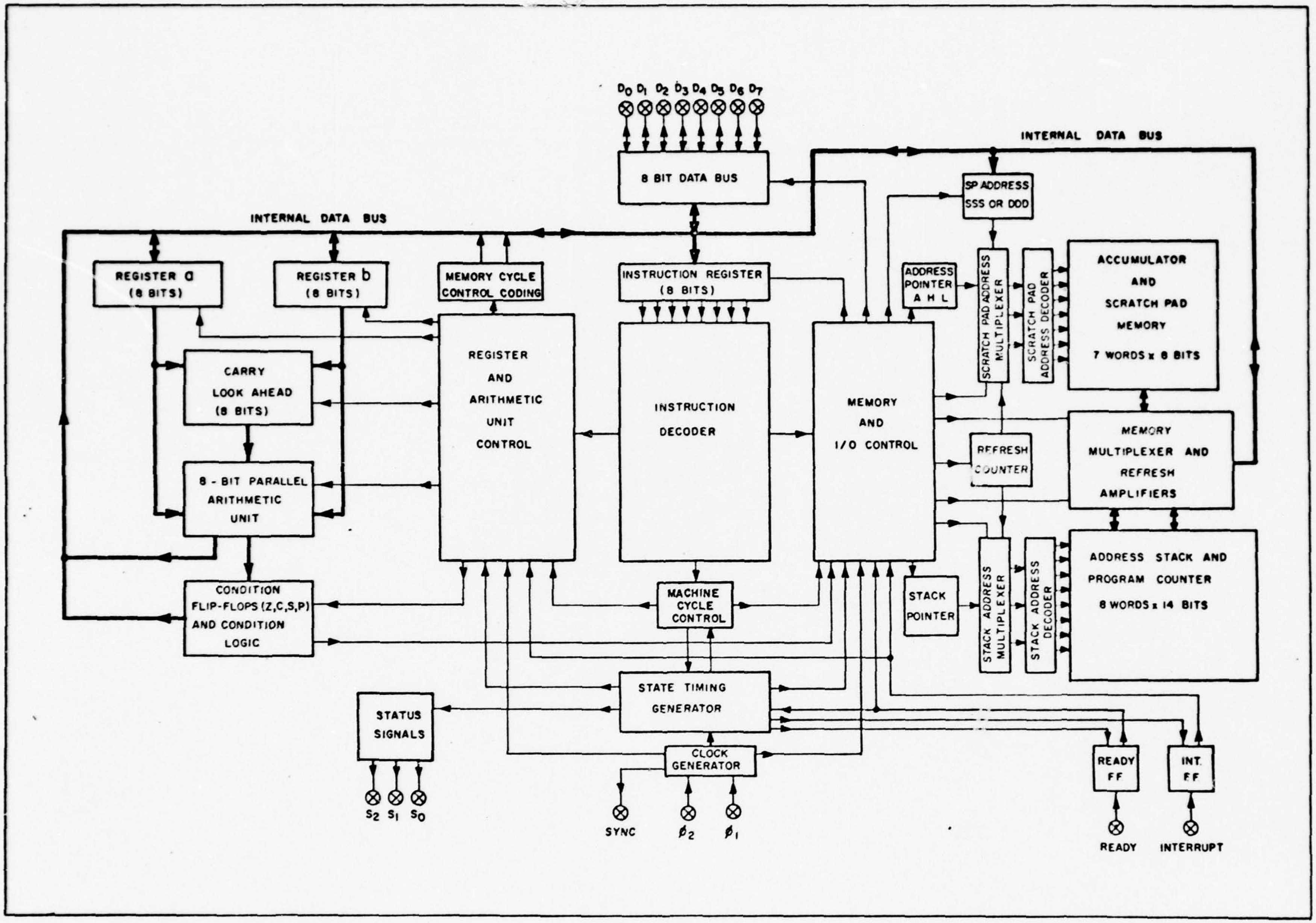
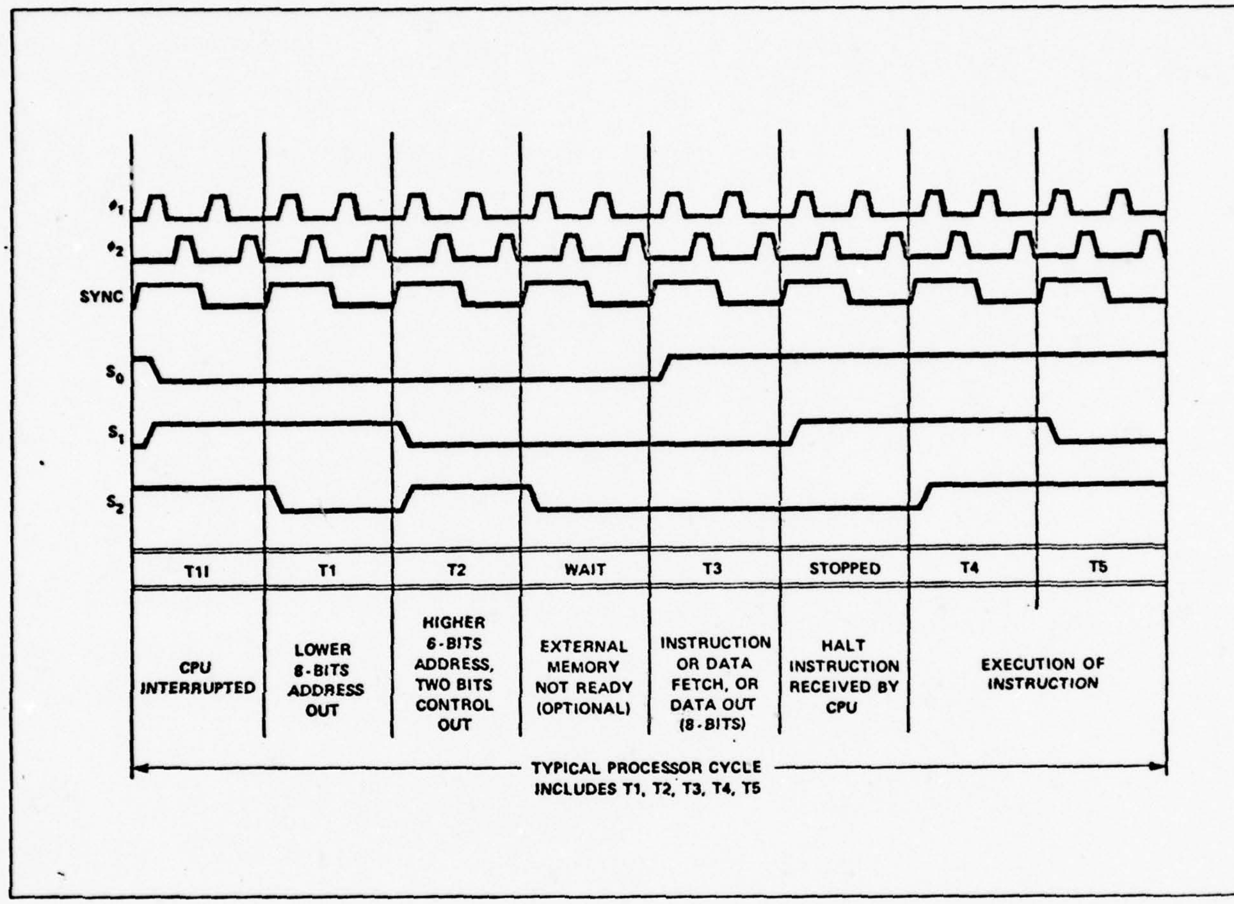


FIGURE 2.26 8008 BLOCK DIAGRAM



BASIC 8008 INSTRUCTION CYLCE

FIGURE 2.27

S_0	S_1	S_2	STATE
0	1	0	T1
0	1	1	T1I
0	0	1	T2
0	0	0	WAIT
1	0	0	T3
1	1	0	STOPPED
1	1	1	T4
1	0	1	T5

MACHINE STATE CODES

TABLE 2.11

D_6	D_7	CYCLE	FUNCTION
0	0	PCI	Designates the address is for a memory read (first byte of instruction).
0	1	PCR	Designates the address is for a memory read data (additional bytes of instruction or data).
1	0	PCC	Designates the data as a command I/O operation.
1	1	PCW	Designates the address is for a memory write data.

CYCLE CONTROL BIT CODES

TABLE 2.12

instruction fetch cycle (PC1). The second and third cycles are for data reading (PCR), data writing (PCW), or I/O operations (PCC).

The cycle types are coded with two bits, D_6 and D_7, and are only present on the data bus during T2. (see Table 2.12)

2.3.2 THE PROCESSOR CONSTRUCTION DRAWINGS

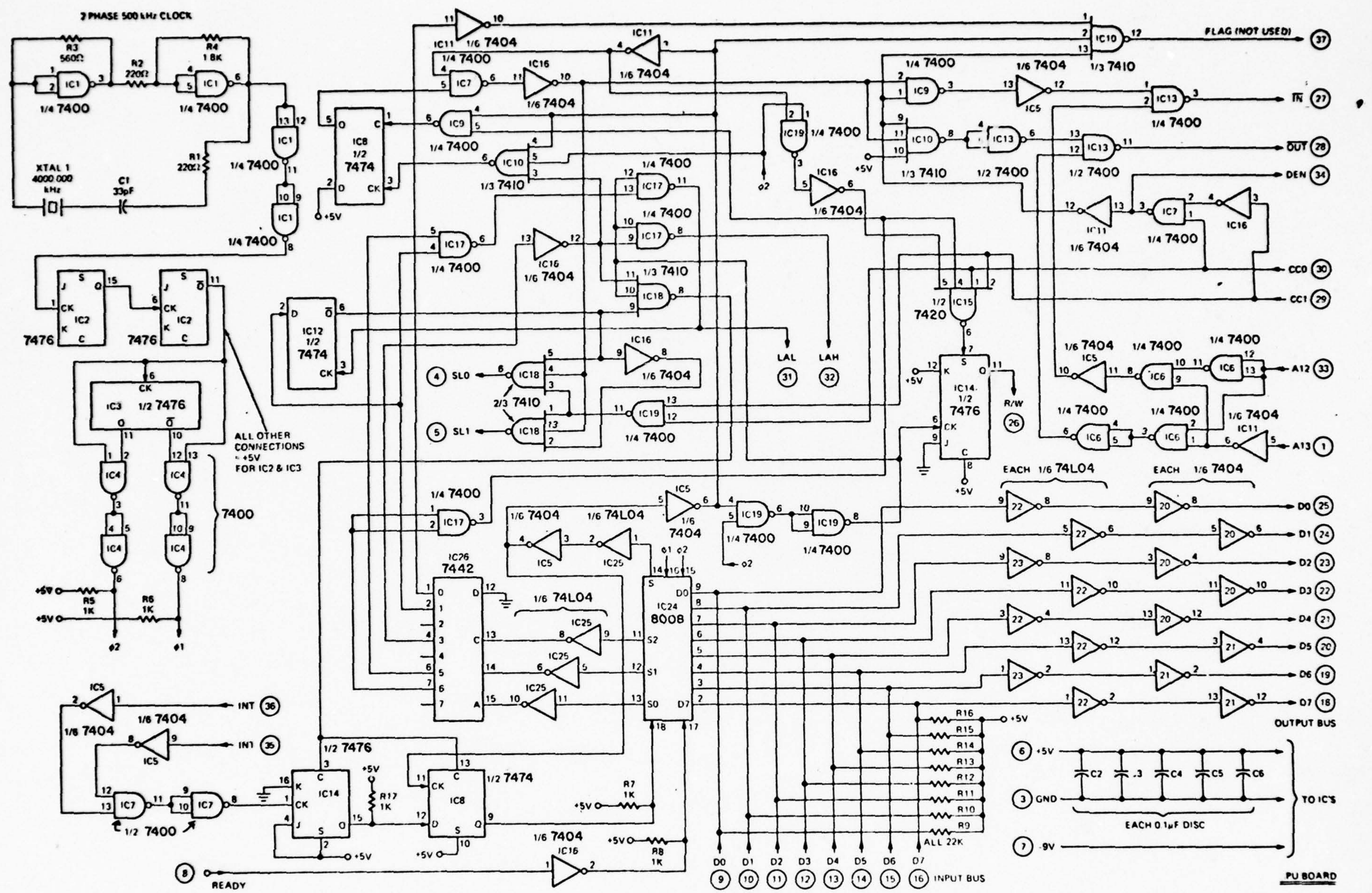
The processor consists of six boards and the front panel switch registers and controls. The six modules are:

- CPU
- Memory address/manual control
- Memory
- Data input multiplexer
- Output latch
- LED register display

Each module is described in the following pages. The descriptions are accompanied by a schematic diagram and a parts layout for each board.

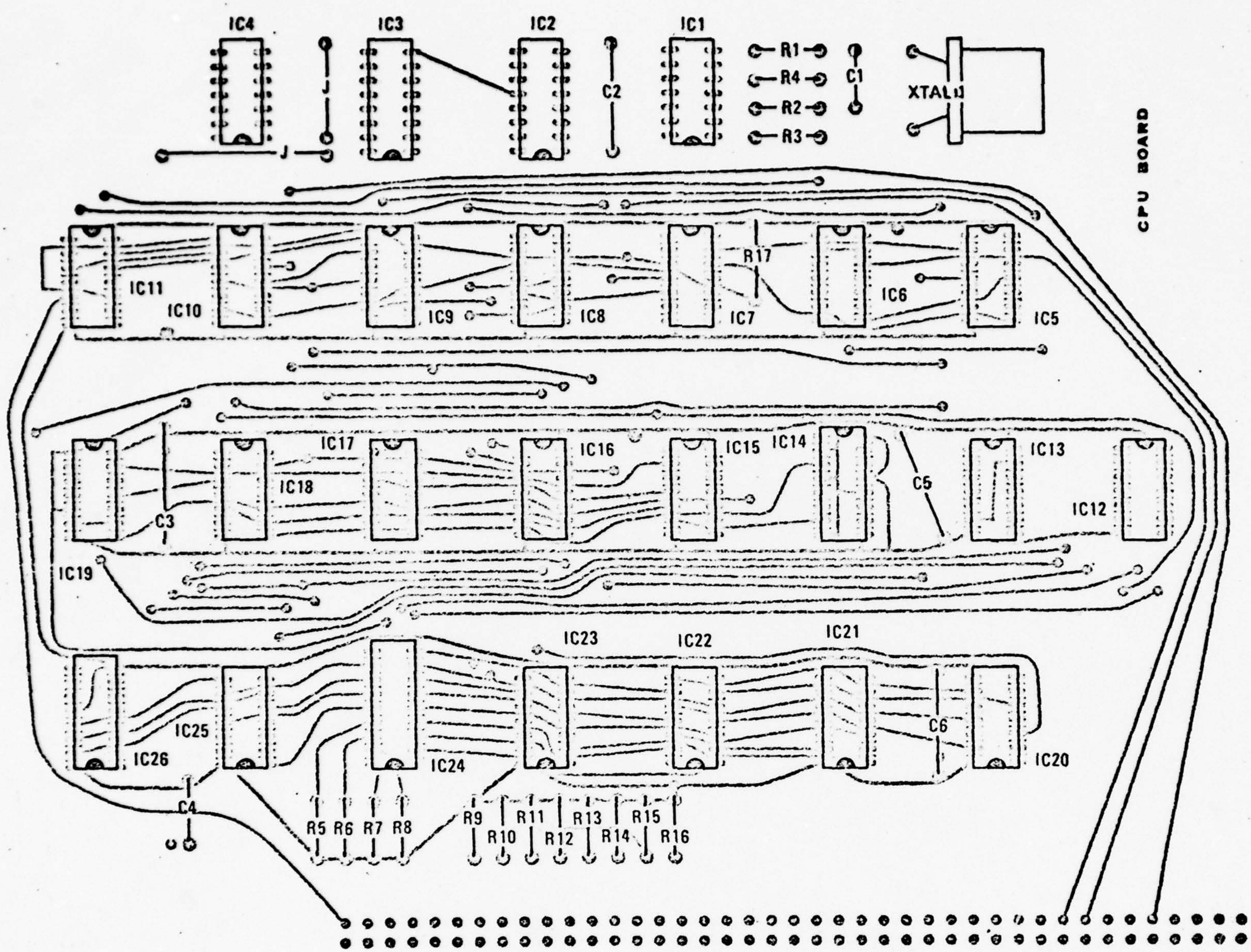
2.3.2.1 CENTRAL PROCESSOR UNIT

The Central Processor Unit (CPU) module (see Figure 2.28 and 2.29) contains the microprocessor IC and the extra circuitry used to interface with the rest of the computer. It is important to note that the 8008 microprocessor is fabricated as an MOS circuit and the outputs will only drive one low-power circuit of the 74L series. Each output is buffered with a 74L04 inverter before it is used. The main, 8-line input/output bus, or I/O bus is also buffered by two 7404 circuits to give the TTL signals a high fan-out.



SCHEMATIC – CPU MODULE

FIGURE 2.28



CPU BOARD - PARTS LAYOUT

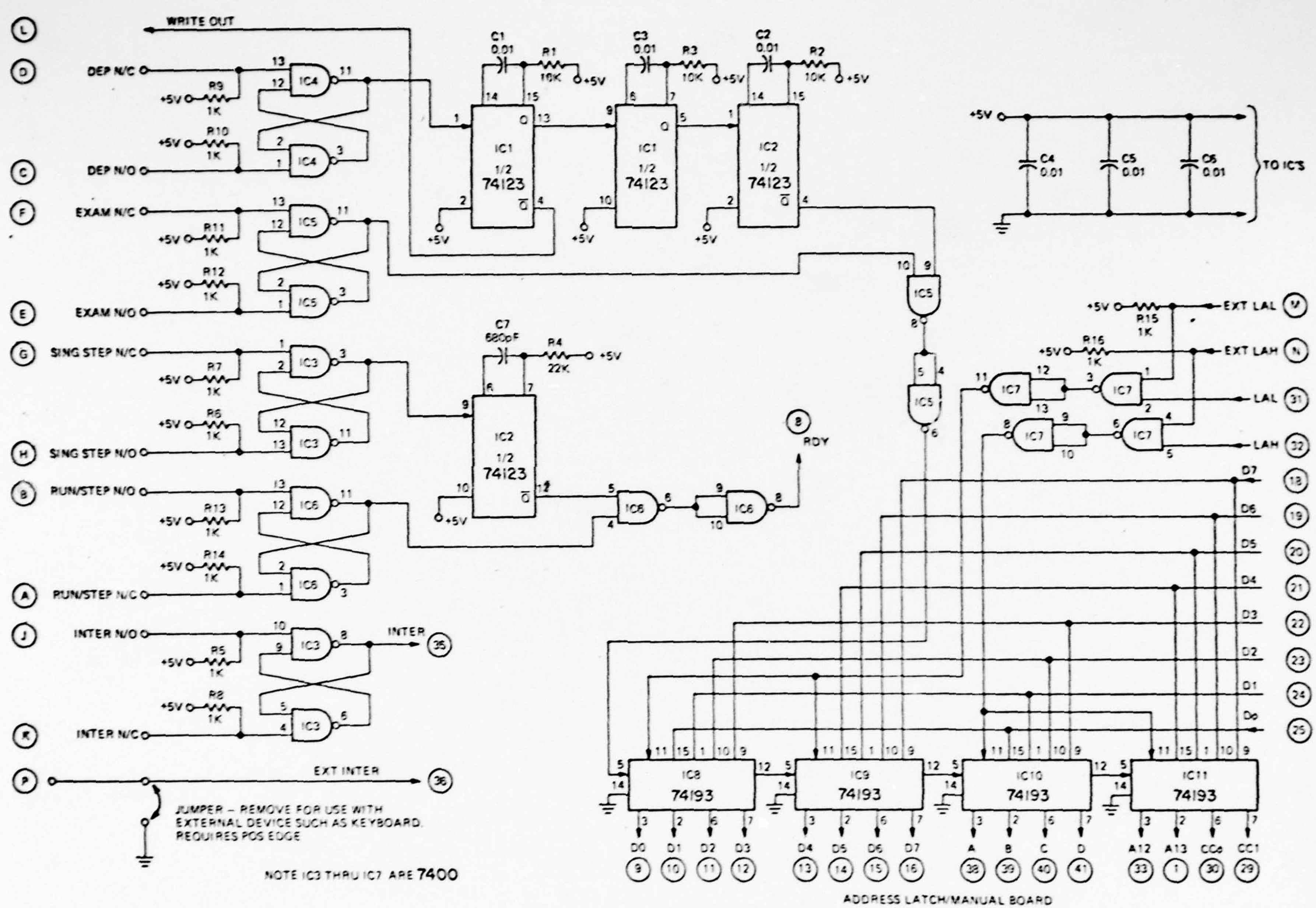
FIGURE 2.29

The computer is controlled by a 2-phase clock supplied by a crystal oscillator which controls the pulse widths and frequency. The clock and the synchronization signal supplied by the microprocessor are used to control some of the logical operations of the computer interface circuits. The synchronization signal synchronizes the operation of TTL circuits and the slower, clocked, MOS circuits in the microprocessor. The microprocessor has three, state-output signals, S_0, S_1, and S_2 which are used to drive a decoder. The eight possible states are then used to control other functions in the interface logic.

Since the CPU uses a parallel 8-bit I/O bus for input and output of data there must be some control of when the bus is sending data from the CPU to an external device or when it is taking data in. Two lines are present on the CPU module, $\overline{\text{IN}}$ and $\overline{\text{OUT}}$. These lines are used by the other modules to regulate the flow of data in the correct direction at the correct time. The control of the IN and OUT lines is governed by the additional logic on the CPU module.

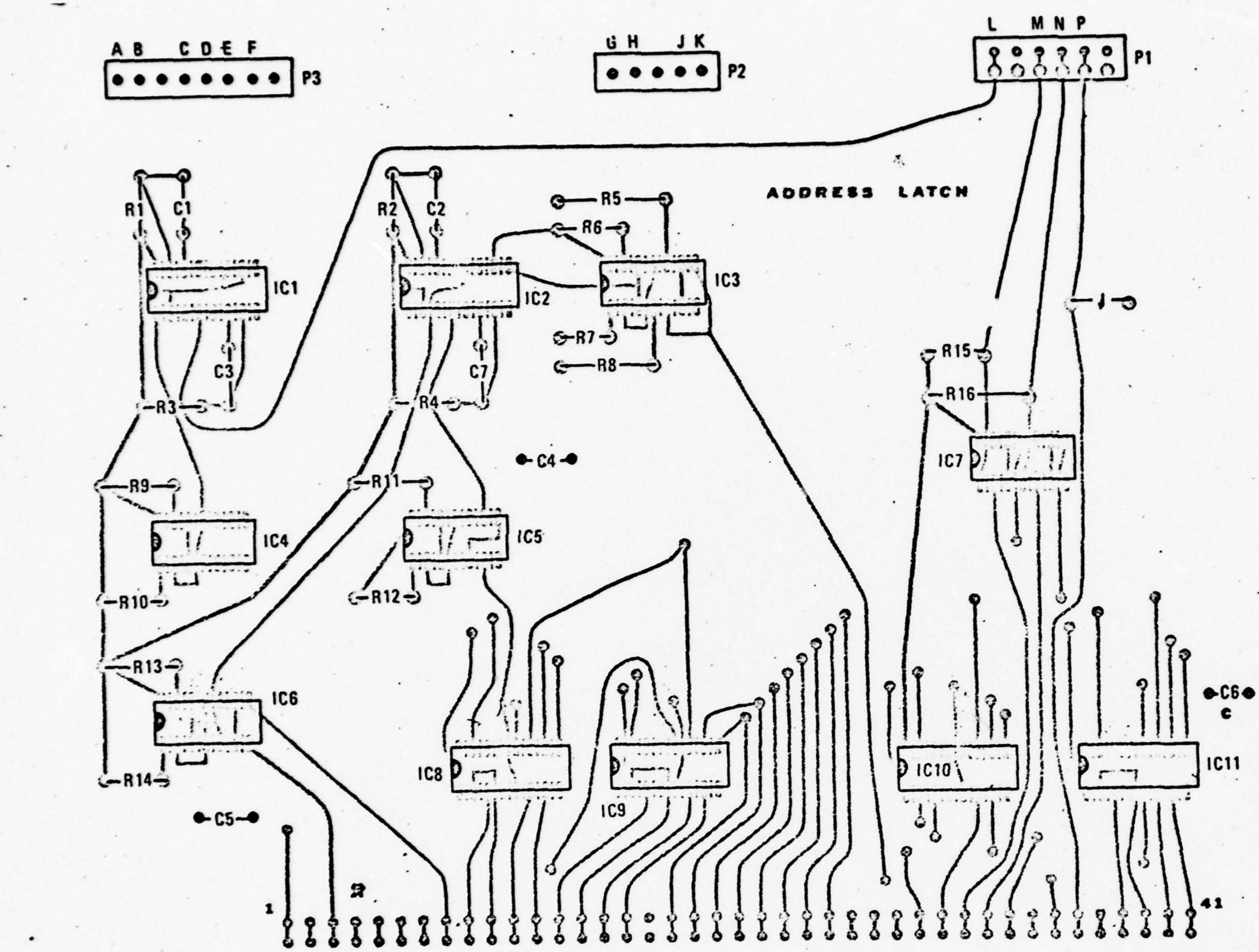
2.3.2.2 MEMORY ADDRESS/MANUAL CONTROL

The Memory Address/Manual Control module (see Figures 2.30 and 3.31) is used to hold data which is to be used as the memory address. Two 8-bit latches are provided since the computer will use one set of eight bits for a memory address and the other set of eight bits for control functions. Since the microprocessor can directly address up to 16,384 words of memory 14 bits are needed for the complete address. The complete memory address of any location is given by a 16-bit number; X X B_3 B_3 B_3 B_3 B_3 B_3/B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2, where X's represent bits that are not used. The computer specifies an address by first sending out the B_2 bits to one of the eight-bit latches, followed by the six B_3 bits and two X bits. Control of the correct latch is supplied from the CPU module.



SCHEMATIC – MEMORY ADDRESS/MANUAL CONTROL MODULE

FIGURE 2.30



ADDRESS LATCH-PARTS LAYOUT
FIGURE 2.31

The B_3 bits are the most significant or the HI part of the address, while the B_2 bits are the least significant or the LO part of the address. Both the HI and LO address latches are made up of SN74193 programmable counters. The HI and LO latches are also used for temporary data storage when they are not being used to store a memory address.

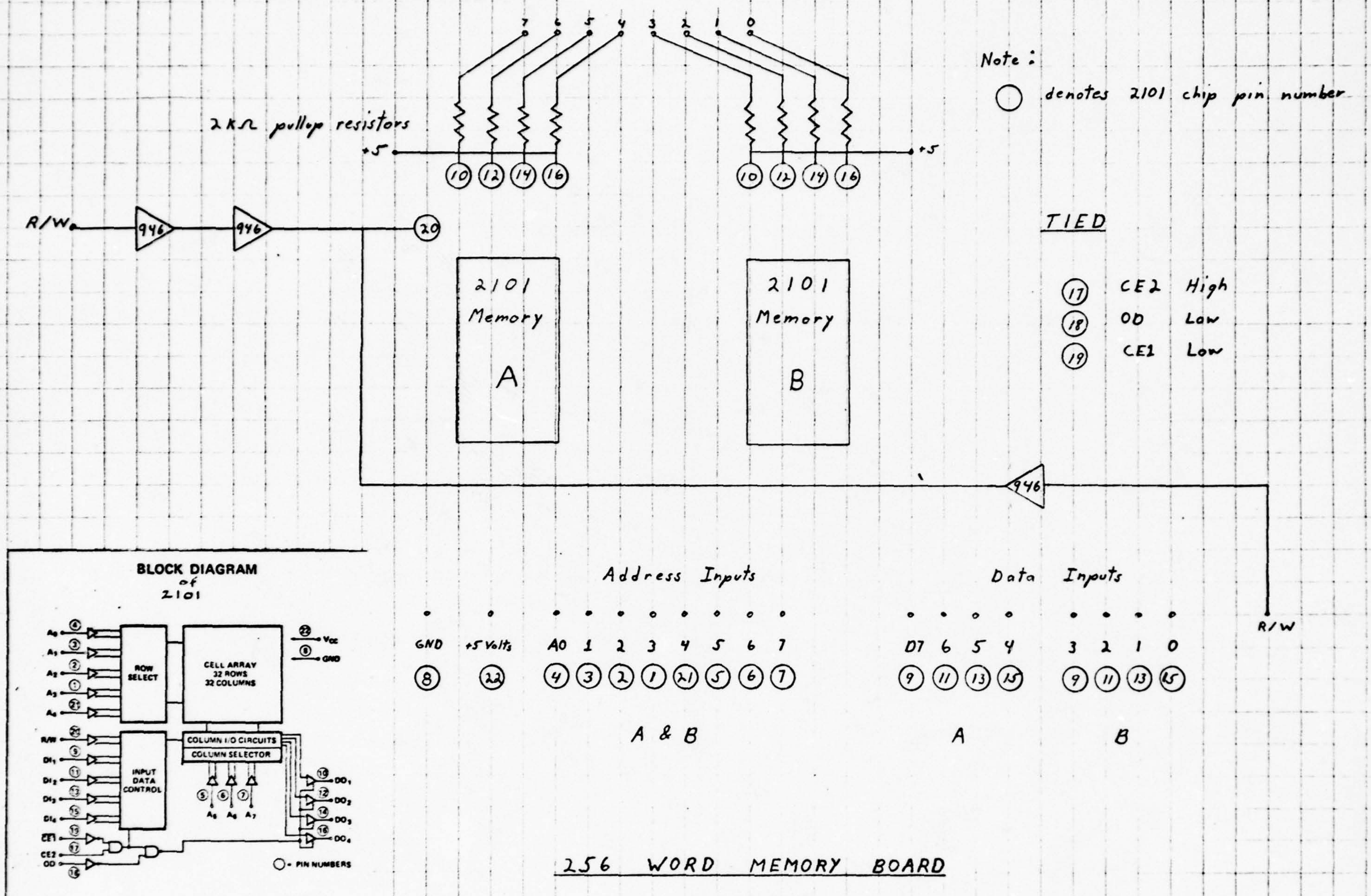
The manual control portion of this module allows programming the computer and controlling its operation from the front panel. It is possible to externally address any memory location and deposit data or instructions in it and also possible to return to any location and check the data stored there. Controls are also provided to allow single-steping the computer through a program, one instruction at a time and to interrupt the computer while it is executing a program.

2.3.2.3 MEMORY MODULE*

The Memory Module (see Figure 2.32) uses the Intel 2101 semiconductor, integrated circuit memory. The 2101 is a static RAM organized as a 256 x 4-bit memory. Two of the 2101 type memories are used to give 256, eight-bit words. Each of the 256_8 words are addressed by the eight bits from the LO address latch. Since 2 = 256 only 256 words are addressable using the LO address alone. Each memory also has an enable line so blocks of 256 words may be selected. The HI address is, therefore, used and decoded with a standard decoder and the decoded outputs are used to enable or select the blocks.

The 2101 type memories are volatile semiconductor memories and information stored in them will be altered or lost if the power is shut off. A read/write or R/W line is provided on the module so that data may either be read from, or written into a selected memory location. The CPU and the Manual Control

*NOTE that the memory board used is not a Techniques Inc. printed circuit board but is wire wrapped on Vector board.



256 WORD MEMORY BOARD

FIGURE 2.32

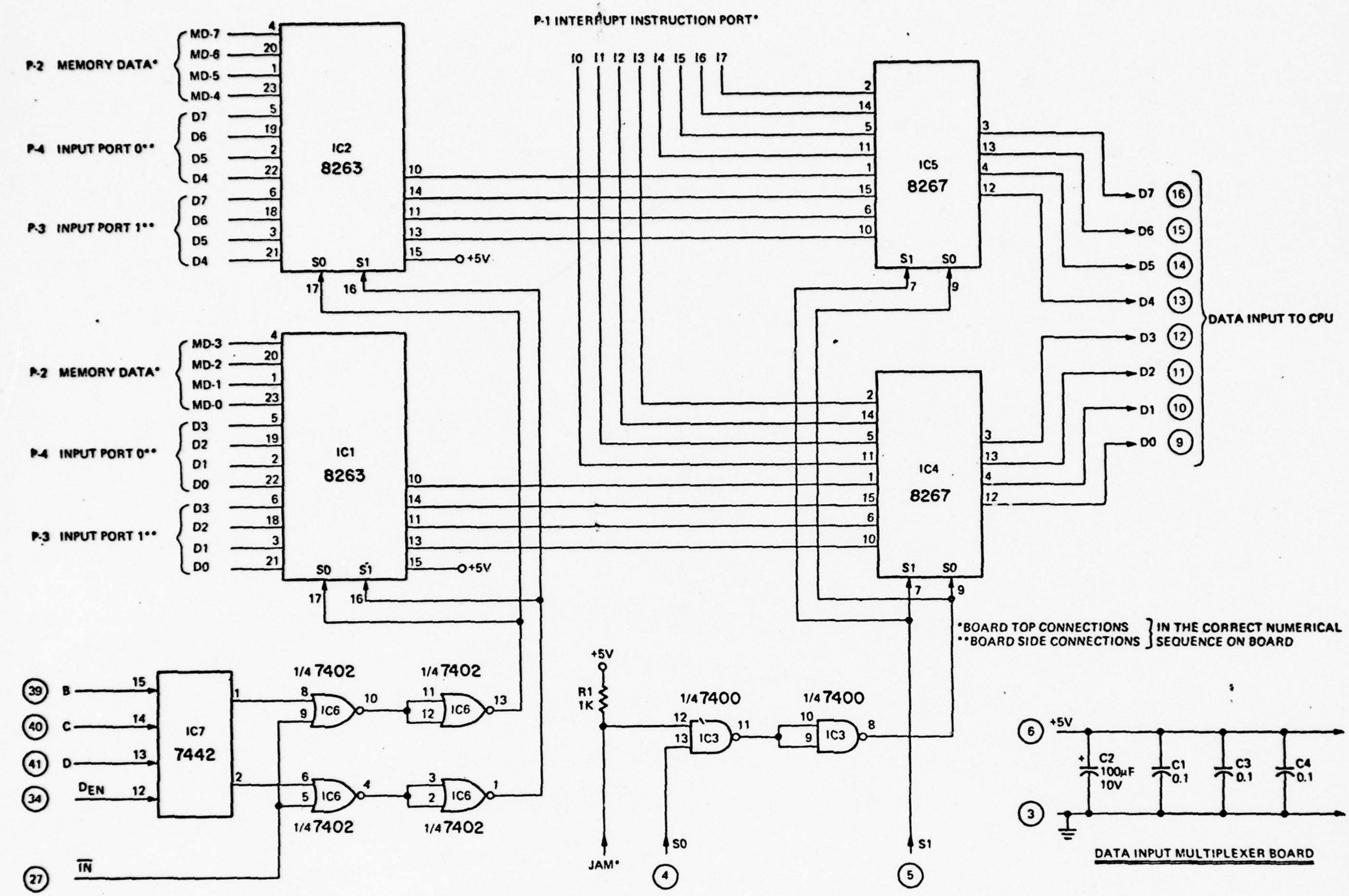
module both control this line so that data may be entered under computer control or so that program data may be entered into the memory prior to use by the computer.

The eight data-output lines from the memory are sent to the CPU I/O bus through the Input Multiplexer module. When accessing data from the memory the CPU senses that the memory data is needed and it sets the input multiplexer so that the data is placed on the I/O bus at the proper time.

2.3.2.4 DATA INPUT MULTIPLEXER MODULE

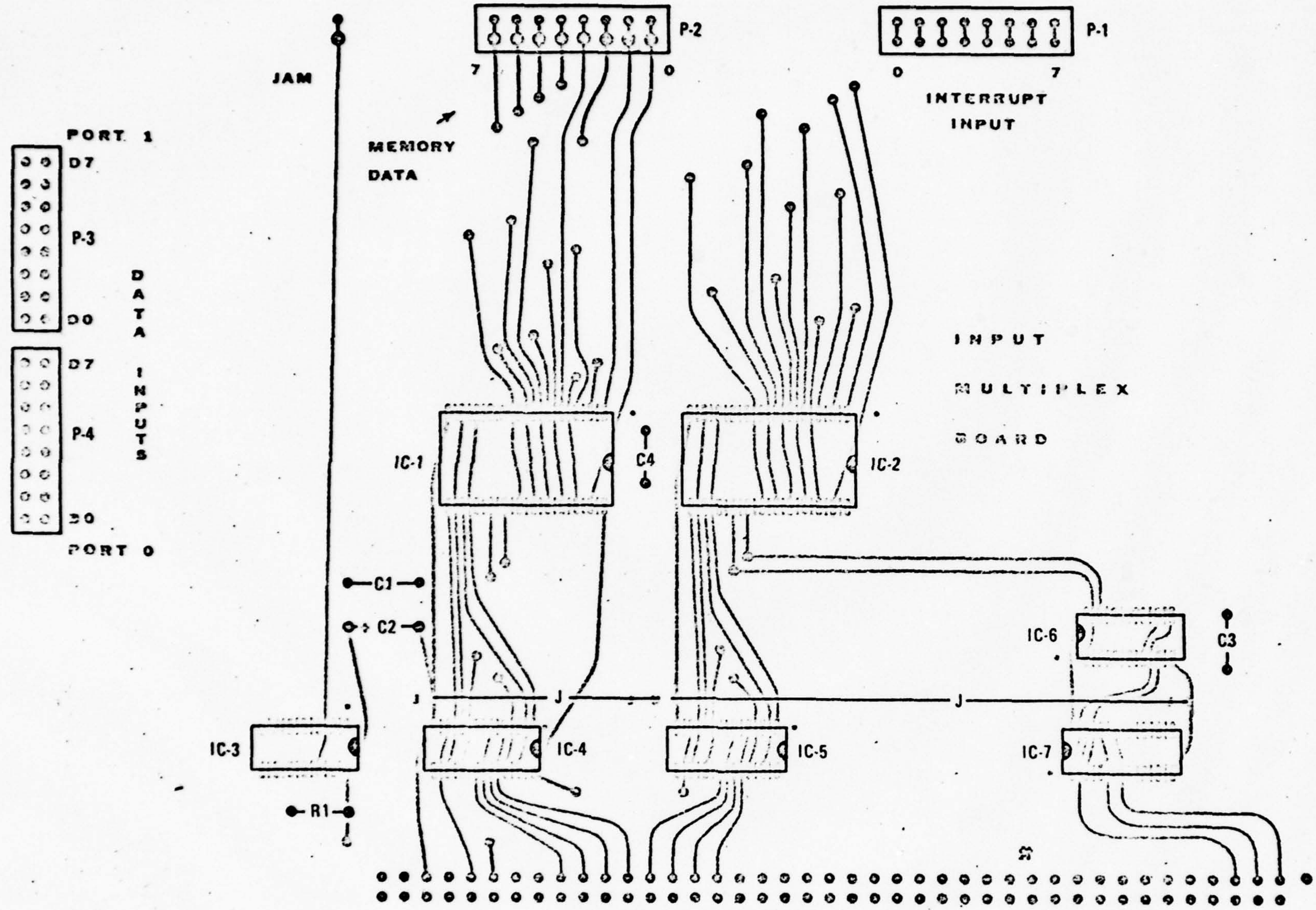
The Data Input Multiplexer module (see Figures 2.33 and 2.34) controls the flow of all data into the computer. All data going into the computer is placed on the I/O bus during the IN cycle signalled by the IN signal. Since data may be coming in from a number of different sources, there must be a means of selecting which data is fed into the CPU. Two basic multiplexers are used for this precise gating of data. The two 8263 quad, three-line to one-line multiplexers control which of three sets of input lines are selected. Two sets of these input lines are input ports 0 and 1. These are the two external data input ports. The third set of data input lines comes from the memory. Data or instructions in the memory, all go through the multiplexer and into the CPU.

This multiplexer is followed by a second set of multiplexers, 8267's. These are quad, two-line to one-line multiplexers with open-collector outputs which are compatible with the computer bus structure. This multiplexer switches between the data selected at the previous multiplexer and data from the Interrupt Instruction Port. This second multiplexer may also be in an off or unselected state which is used when data is not to be sent to the CPU module. Control lines SL_0 and SL_1 are sent directly from the CPU interface logic.



SCHEMATIC – DATA INPUT MULTIPLEXER MODULE

FIGURE 2. 33



DATA INPUT MPX BOARD - PARTS LAYOUT
FIGURE 2.34

When the HI address is not being used to store a memory address, it is used for control signals. During an IN or OUT cycle these control signals are decoded and used to select the proper input or output lines for the I/O bus. The Multiplexer module decodes the control bits B, C, D, and D_{Enable} and OR's them with IN to select the proper external data input port. When the computer is instructed to get data from memory it automatically selects the memory input section of the multiplexer. The INPUT instruction is used only when data is accessed from some external source.

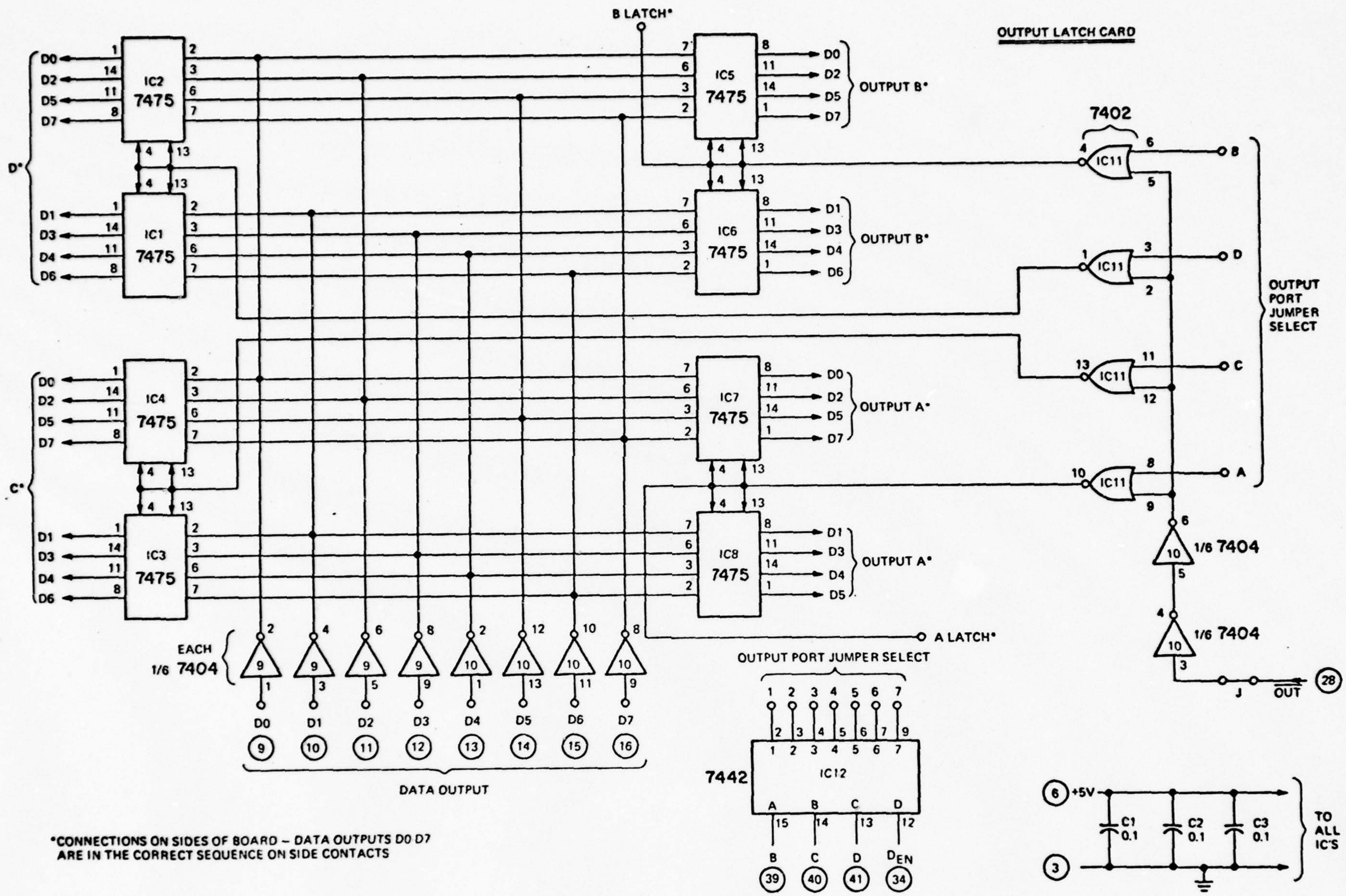
2.3.2.5 OUTPUT LATCH MODULE

The Output Latch Module (see Figures 2.35 and 2.36) is used to send data from the computer to some external device or instrument. Four output latches are provided on the Output Latch Module.

Note that data is sent from the LO address latch to each output port and that these connections are in parallel. The computer decides which latch is activated according to the OUTPUT instruction that is in this program. Here, again, the HI address latch holds the control bits B, C, and D, which are decoded and NORed with OUT to activate the selected eight bit output port or latch. NOTE: The OUTPUT instruction in the Intel User's Manual has two RR bits shown in it. These bits must be set to RR=01 for proper data output. OUT=01 01MMM1. The MMM bits are set to the binary equivalent of the decoder state selected for that particular output port. For example 01 010 111 would output port 3, since 011=MMM=3.

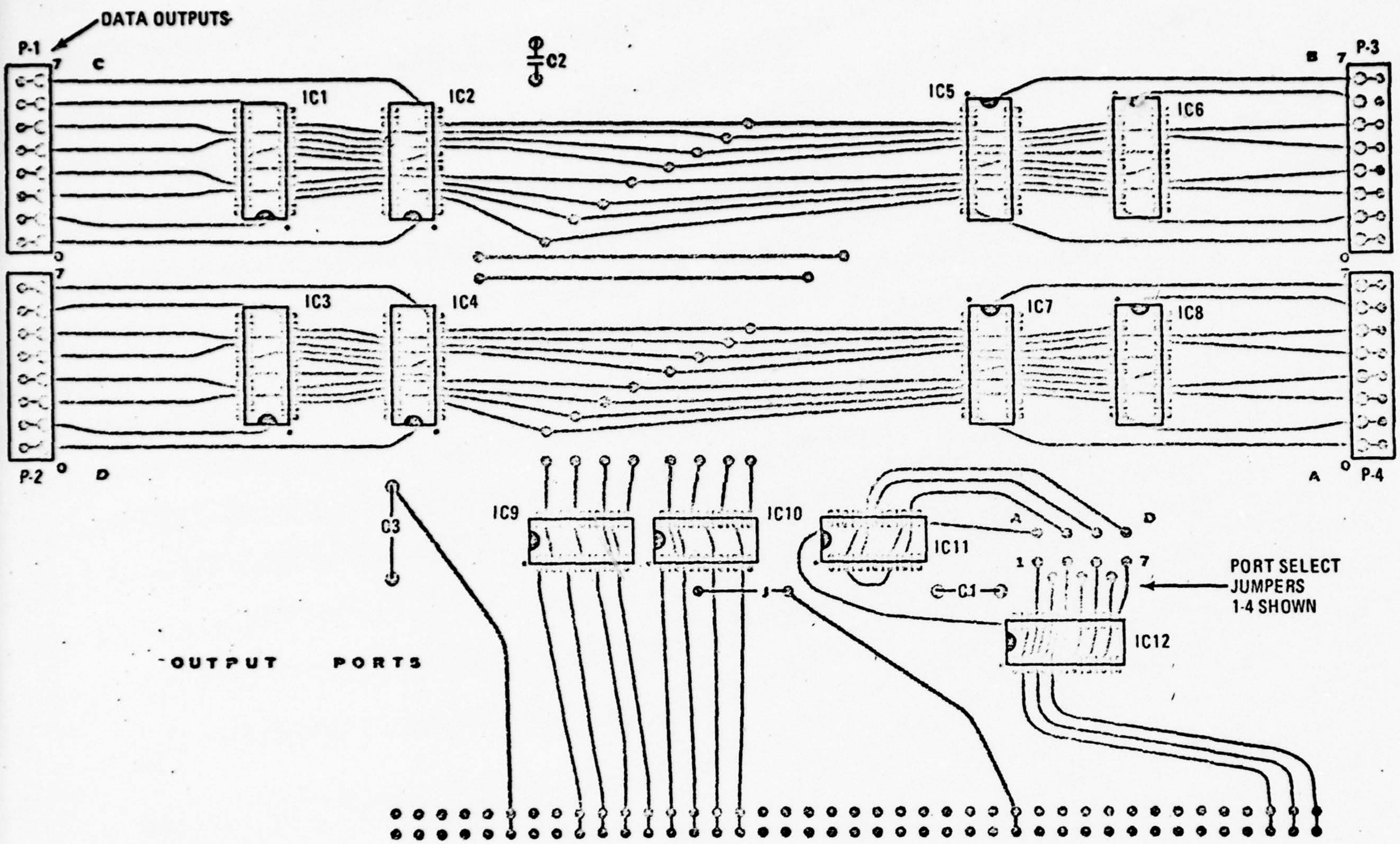
2.3.2.6 LED REGISTER DISPLAY

The LED Register Display module provides a visual indication of the contents of the HI and LO address latches and the memory data in the selected location indicated by that address. Output port 0 is also located on the readout



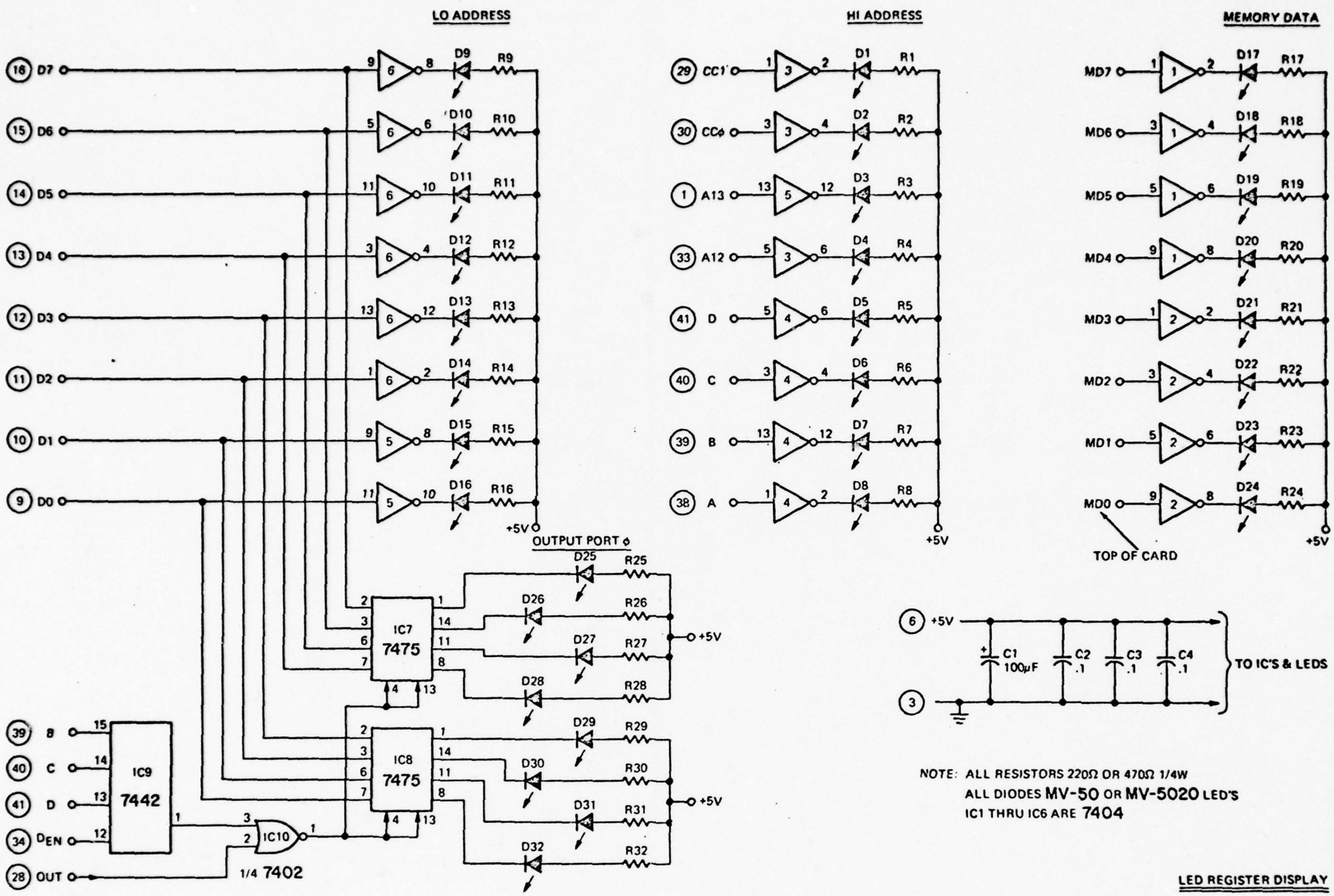
SCHEMATIC – OUTPUT LATCH MODULE

FIGURE 2.35



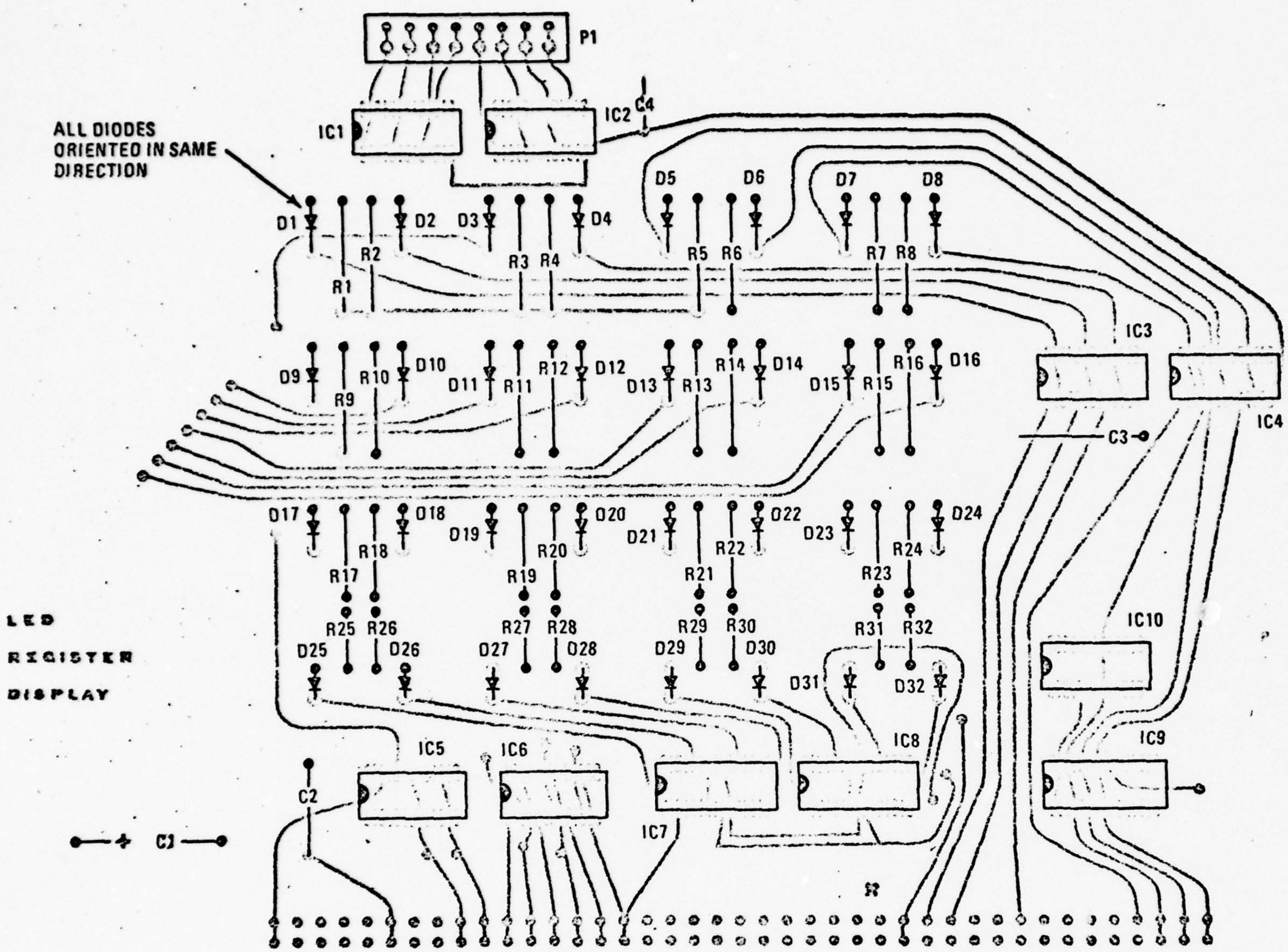
OUTPUT PORTS-PARTS LAYOUT

FIGURE 2. 36



SCHEMATIC – LED REGISTER DISPLAY

FIGURE 2.37



LED REGISTER DISPLAY-PARTS LAYOUT

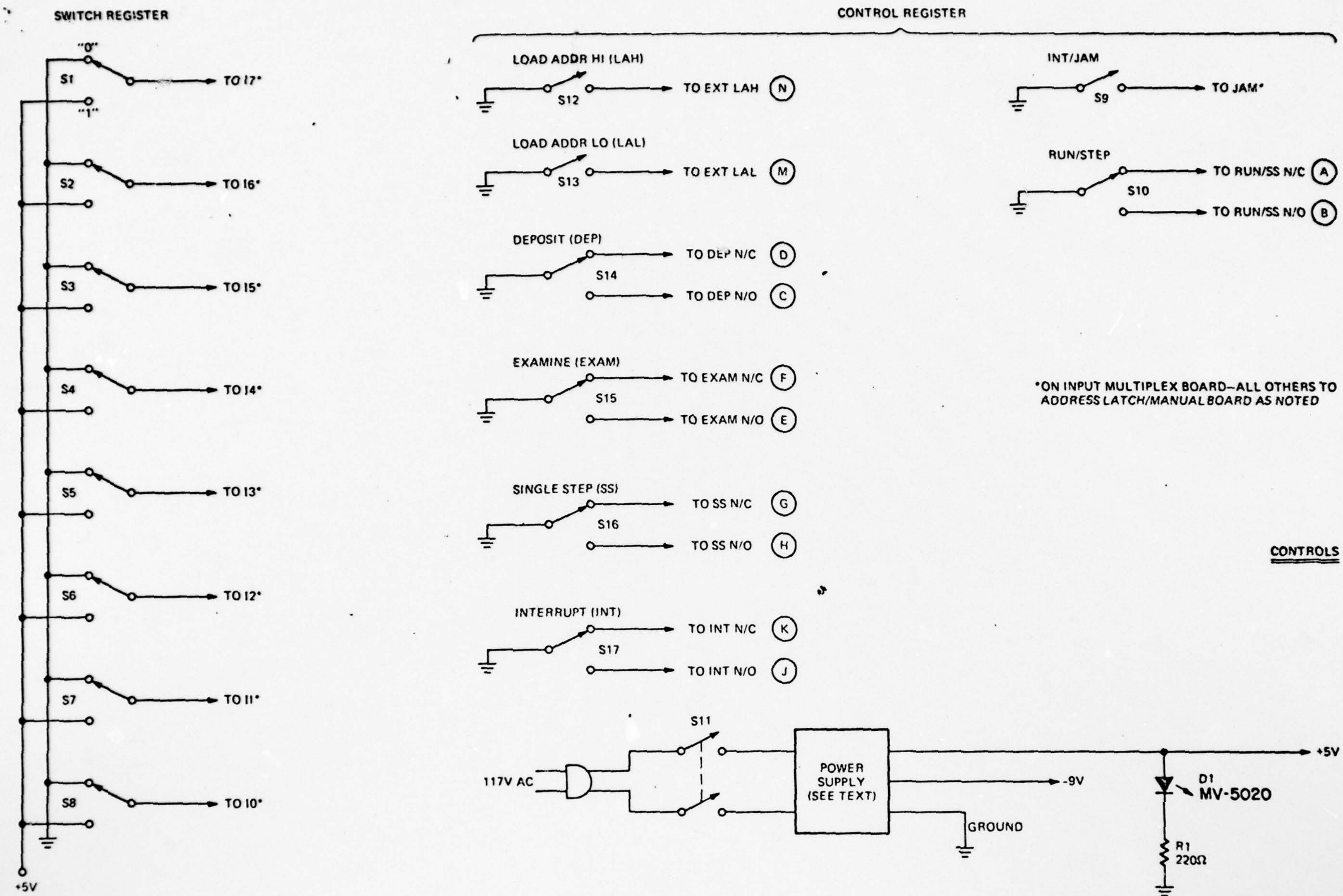
FIGURE 2. 38

module and it may be used in programming to give a visual output of a byte of data. Each of the output registers is represented by eight LED indicators, 1=ON, 0=OFF. As the data held in each register changes, so do the indicators. Data to be displayed at outport 0 must be sent with an OUT instruction 01 010 001 or 121_8.

Since the HI address latch is used for some control functions and the LO address latch may also be used for temporary storage of data going to the output ports, at various times in programs the data in these registers will change from a memory address to these control and output data and then back to an address. Checking this data visually in these registers during the debugging of a program is very helpful.

2.3.3 CONTROLS

Eight switches are provided on the Interrupt Instruction Port (see Figure 2.39). This is called the Switch Register or SR (located on front panel) and it is one way to get data into the computer under manual control. Notice on the Input Multiplexer module schematic diagram that the SL_0 signal is gated with a Jam signal. When the Jam is at ground, this forces the SL_0 signal to also go to ground. When this happens, the 8267 multiplexers are held in the state which allows the data present at the Interrupt Instruction Port to be placed on the I/O bus, going to the memory and to the HI and LO address latches. When the INTERRUPT/JAM switch is returned to the normal INTERRUPT position, control of the SL_0 line is taken over by the CPU control logic. The JAM control allows jamming data onto the I/O bus. This Jam mode is useful only when the computer is not operating, but before starting the computer the INTERRUPT/JAM control switch must be in the normal INTERRUPT position.



SCHEMATIC – SWITCH REGISTER

FIGURE 2.39

This program starts at location 00 000 and can be used to test dynamic operation of the computer.

ADDRESS	MNEMOMIC	OCTAL	COMMENTS
00 000	LDCI	026	Load C immediate
00 001		000	Data to be loaded
00 002	LDAI	006	Load A immediate
00 003		000	Data to be loaded
00 004	LDBA	310	Transfer A to B
00 005	INCB	010	Increment B by +1
00 006	LDAB	301	Transfer B to A
00 007	OUT	121	Output A to port 0
00 010	LDBA	310	Return A to B
00 011	LDAC	302	Transfer C to A
00 012	ADDI	004	Add immediate
00 013		001	Data to be added
00 014	JPTC	140	Jump if Carry is True
00 015		005	LO Address of jump
00 016		000	HI Address of jump
00 017	JPUN	104	Unconditional jump
00 020		012	LO Address of jump
00 021		000	HI Address of jump

TABLE 2.13

REGISTER INCREMENT TEST PROGRAM

The computer is normally in the RUN state and it only halts when it reaches a halt or HLT type instruction in the program. To see how a program works at slow speed it is necessary to slow down the computer. The computer has a RUN/SINGLE STEP switch which allows one to either run the program at the normal computer speed of 50,000 steps per second, or at a step at a time. In the RUN mode the computer operates at its own speed, determined by the clock. In the SINGLE STEP mode the computer is pulsed each time the SINGLE STEP switch is pressed causing a complete computer cycle to take place.

The use of the next four controls allows entering data into the computer and checking data already stored in memory, before starting the computer program. When using any one of these next four controls, the INTERRUPT/JAM switch must be in the JAM position and the RUN/SINGLE STEP switch must be in the SINGLE STEP position.

The LOAD ADDRESS-HI or LAH switch allows loading the HI address latch with the number currently set on the switches in the switch register. The LOAD ADDRESS-LO or LAL switch is operated in exactly the same way, entering the number set on the switch register to the LO address latch. Using the eight switch register switches and these two controls one can manually load any address in the address latches. The new address appears on the LED indicators in the HI and LO address readouts. As soon as an address has been loaded the memory data LEDs will indicate the current contents of the location just addressed. All possible 16,000 storage locations in the memory from the switch register can be addressed but is important to note that it is only realistic to try and address memory locations that actually exist as implemented (i.e. locations 000 to 256).

The DEPOSIT switch allows depositing data from the switch register to the memory location addressed. Once the HI and LO addresses have been loaded and checked, set the switch register to the value of the data to be entered into

that location. Pressing DEPOSIT causes the computer to write the data word set in the switch register in the memory location selected. The memory address is automatically incremented when the DEPOSIT switch is actuated. This is done by using the SN74193 programmable counters as the address latches. By automatically incrementing the address the next memory location is loaded without having to reload the next successive memory address. Data may now be deposited in the next memory location and the next and so on just by setting the data in the switch register and actuating DEPOSIT. The address steps to the next location automatically. In this way blocks of data are easily stored in successive locations.

Another control, EXAMINE, is also provided. Once an address is loaded in the HI and LO address latches using LAH and LAL the contents of that location are displayed in the memory data LED readouts. Actuating EXAMINE steps the memory address to the next location without altering the data stored there. Consecutive locations may be examined just by depressing the EXAMINE switch. This allows checking programs or data without altering the data present.

The INTERRUPT is used to interrupt an executing program and cause the computer to temporarily do some other task. It is also possible to make the computer leave a stopped state with the interrupt.

Any instruction may be set on the switch register to be fed into the Interrupt Instruction Port when the computer is interrupted. Multiple part instructions such as the unconditional jump instruction could also be entered, but it would take some extra interface logic to do this. Usually only signal byte instructions are entered. The 300 instruction, continue or no-operation, a halt or HLT and the restart or RST instructions are usually the only ones entered while the computer is running at its normal speed.

The Restart or RST Instruction is one of the most useful instructions that may be entered from the switch register when interrupting the computer. It is used to start programs and it is extremely useful when using the computer with external devices. The Restart instruction is a pointer type of instruction that points the computer to a particular location without a multiple byte jump instruction. The RST instruction has three bits labeled A 00 AAA 101. The A's are set to the starting address of the interrupt program and all other bits in the address are zero; HI=00 000 000, LO=00 AAA 000. Note that only eight particular locations may be accessed with the RST instruction (i.e. 000, 010, 020, 030, 040, 050, 060, 070). Putting a RST instruction on the switch register and hitting the interrupt switch will cause the computer to begin executing the program starting at location AAA.

The following sample program (Table 2.13) when loaded and executed will cause the output 0 display on the front panel to increment. By changing step 00007 to Out 1 or Out 2 (123_8 and 125_8 respectively) and selecting the discrete output display on the front panel the high or low order discrete output bits may be made to increment.

2.3.4 THE 8008 INSTRUCTION SET

Appendix F contains material describing the 8008 instruction set. It is reproduced directly from the Intel 8008 Users Manual.

3.0 STINGER REAL-TIME/IRSS TARGET GENERATION

3.0 STINGER REAL-TIME/IRSS TARGET GENERATION

Documentation of the real time STINGER/IRSS target generation equations is given in this and the following section. The documentation given in these two sections provides information concerning the existing simulation with regard to two primary areas:

- Specific mathematical equations currently implemented for target generation
- Input/output of real time information

In total the collection of information presented in the following sections, provides a description of the mathematical model of target generation up to the point of specific computer implementation. The discussion of real time I/O is intended to clarify the logical flow of real time information between the simulation elements and to provide a basis for future partitioning efforts.

Some information has been deleted in view of other documentation efforts currently in preparation. Specifically, the information not contained here includes:

- Rationale of the mathematical models
- Specific analog and digital computer implementations

In addition to the material previously mentioned there are five Appendices which include related material. Appendix A contains IRSS calibration limits and sign conventions. Appendix B, C, and D contain functional operation sequences for the three major simulation control centers. And, Appendix E addresses in a general context characteristics of the direct cell. These Appendices are intended to offer alternate perspectives (though individually limited) of the STINGER real time simulation.

3.1 TARGET GENERATION EQUATIONS

In this section the mathematical equations which currently describe the target generation in the STINGER real time/IRSS simulation are listed. These lists are organized into logical sequences which lead to physical variables. Specifically the list of equations summarizes the current implementation and order of computation for:

- Plume shape
- Plume azimuth and elevation
- Plume length to breadth ratio and aspect angle
- Plume iris ratio
- Plume rotation angle
- Target angle of attack and mach number
- Flare transparency radius and iris ratio
- Flare azimuth and elevation
- Flare aerodynamics
- Tailpipe azimuth and elevation
- Coordinate transformations
- Variable scale factors

This list of equations is supplemented by Figures 3.1 through 3.6 and Table 3.1. Figures 3.1 through 3.6 provide a graphical basis for interpreting the mathematical notation. These figures are not intended to provide a rationale for the equations presented, but rather, are presented in order to provide a quick reference of physical relationships. The equations are given further clarity when considered in conjunction with Table 3.1. Table 3.1 is a cross reference of physical variables, mathematical notation and notations used in the various computer codes (both analog and digital).

In Figure 3.7 a flow chart of the task currently performed by the CDC/6600 is given. This flow chart provides a view of the real time simulation from one perspective. With the aid of Table 3.1 and the I/O description given in section 3.2 the nomerclature encountered in the flow chart is explained.

PLUME SHAPE

$$|\bar{L}_{os}| = \sqrt{(X_{ML}-X_{TL})^2 + (Y_{ML}-Y_{TL})^2 + (Z_{ML}-Z_{TL})^2}$$

$$P_1 = 2\tan^{-1}\left(\frac{B/2}{\ell}\right) \qquad \text{(DEG)}$$

$$P_2\text{INCH} = \frac{(f)(\ell)}{|\bar{L}_{os}|} \qquad \text{(IN)}$$

$$P_2\text{RAD} = \tan^{-1}(\ell/|\bar{L}_{os}| \qquad \text{(RAD)}$$

$$P_2\text{RAD} \simeq \ell/|\bar{L}_{os}| \qquad \text{(RAD)}$$

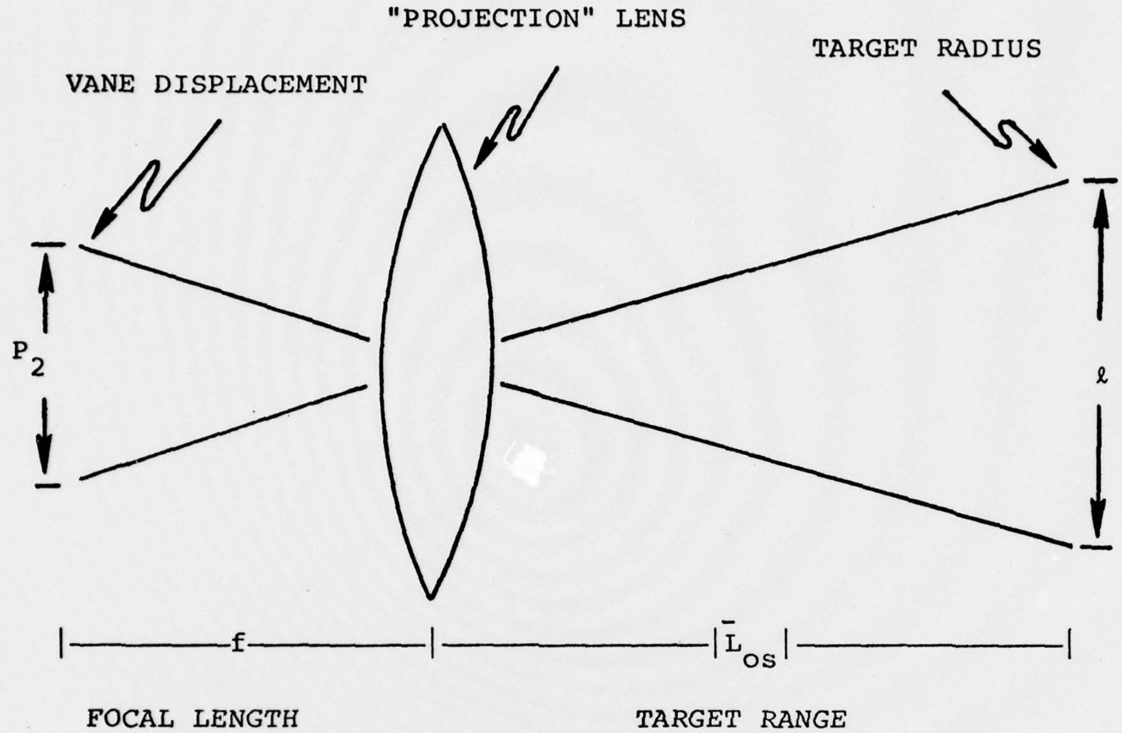


FIGURE 3.1

PLUME TRANSPARENCY

PLUME AZIMUTH AND ELEVATION

$\psi_7 = \psi_4 - P_2 \cos\,(T_{RP})$ (DEG)

$\dot{\psi}_7 = \psi_4$ (DEG/SEC)

$\theta_7 = \theta_4 - P_2 \sin\,(T_{RP})$ (DEG)

$\dot{\theta}_7 = \dot{\theta}_4$ (DEG/SEC)

PLUME LENGTH TO BREADTH RATIO AND ASPECT ANGLE

$$\ell \approx L \sin(\varepsilon) \quad \text{if } B<L,\ B=3.0$$

$$\ell = L \sqrt{1-\cos^2(\varepsilon)}$$

$$\ell/B = \frac{L}{B} \sqrt{1-\cos^2(\varepsilon)}$$

$$\cos(\varepsilon) = \frac{\bar{L}_{os} \cdot \bar{C}_L}{|\bar{L}_{os}|\,|\bar{C}_L|}$$

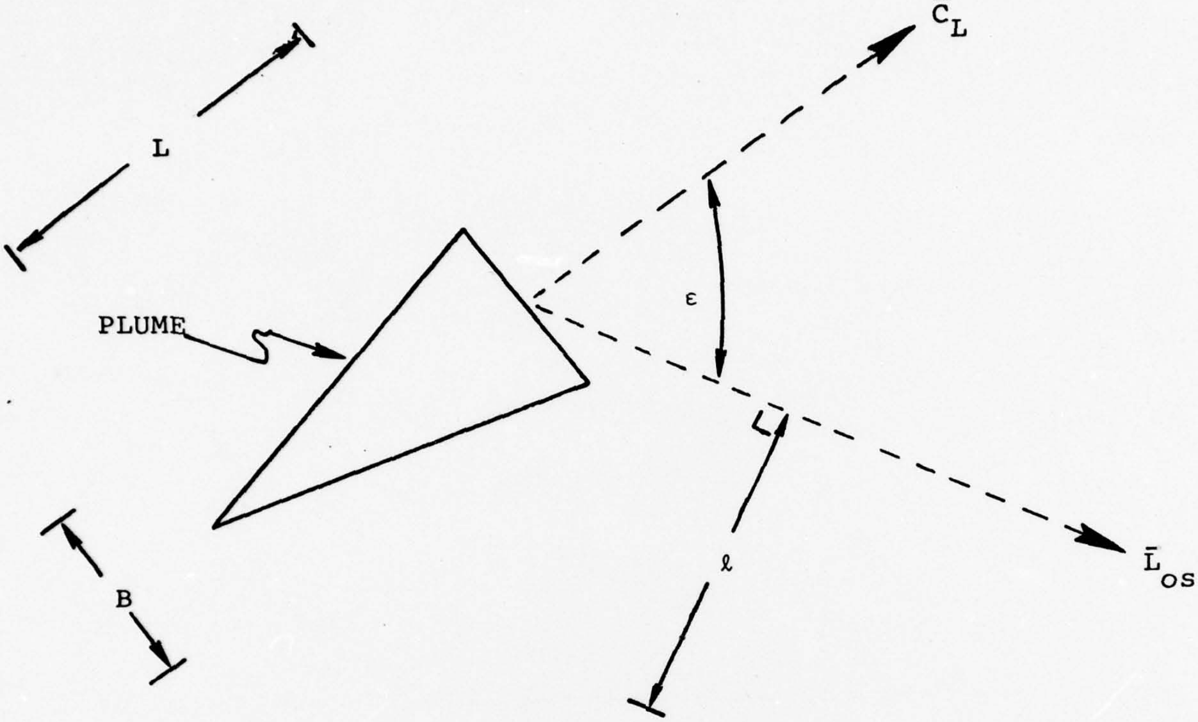


FIGURE 3.2

PLUME APPARENT LENGTH AND ASPECT ANGLE

PLUME IRIS RATIO

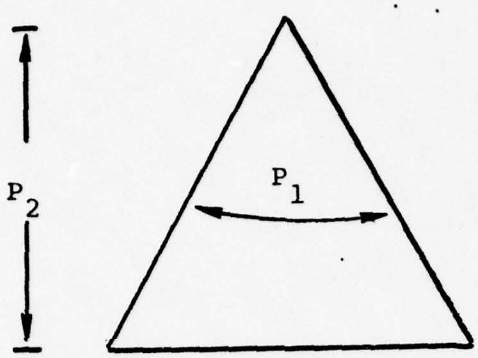
$$H_{c7} = \left(\frac{R}{7000}\right)^{-2.341} \left(H\Big|_{\substack{R=7000 \\ \varepsilon}}\right) \qquad (W/CM^2)$$

$$A_{t7} = \frac{P^2_{2INCH}\ SIN(P_1)}{2\ COS^2\ (P_1/2)} \qquad (IN^2)$$

$$J_{tu} = K_2 A_{t7}^{K_3} \qquad (W/STER)$$

$$I_{r7} = \frac{\dfrac{H_{c7} f^2}{T_{f,plume}}}{J_{tu}}$$

$$i_7 = i_7 \, (I_{r7}) \qquad (TABLE\ LOOK\ UP)$$



$$\boxed{\text{AREA} = P_2^2 \tan (P_1/2)}$$

OR

$$\boxed{\text{AREA} = \frac{P_2^2 \sin P_1}{2 \cos^2 (P_1/2)}}$$

FIGURE 3.3

AREA OF PLUME TRANSPARENCY

AD-A052 680 B-K DYNAMICS INC ROCKVILLE MD F/G 9/2
QUARTERLY INTERIM TECHNICAL REPORT (4TH), CONTRACT DAAH01-75-C---ETC(U)
OCT 75 DAAH01-75-C-0194

UNCLASSIFIED BKD-TR-3-201 NL

2 OF 2
AD
A052680

END
DATE
FILMED
5 -78
DDC

PLUME ROTATION ANGLE

$$\beta = \cos^{-1}\left(\frac{L_x}{\sqrt{L_x^2 + L_y^2}}\right)$$

$$C_x = -\cos\beta$$

$$C_y = \begin{cases} +\sin\beta \text{ if } L_x > 0 \\ -\sin\beta \text{ if } L_x < 0 \end{cases}$$

$$D_X = L_y \cdot C_{LZ} - L_Z \cdot C_{LY}$$

$$D_Y = L_Z \cdot C_{LY} - L_X \cdot C_{LZ}$$

$$D_Z = L_X \cdot C_{LY} - L_Y \cdot C_{LX}$$

$$\cos\theta = \frac{(\bar{L}_{os} \times \bar{C}_L) \cdot \bar{C}}{|\bar{L}_{os} \times \bar{C}_L| \cdot |\bar{C}|}, \text{ or } \theta = \cos^{-1}\left(\frac{(D_X \cdot C_X + D_Y \cdot C_Y)}{\sqrt{D_X^2 + D_Y^2 + D_Z^2}}\right)$$

$$T_{RP} = \begin{cases} \pi/2 - \theta & \text{if } D_Z < 0 \\ \pi/2 + \theta & \text{if } D_Z > 0 \text{ and } \theta < \pi/2 \\ \theta - 3\pi/2 & \text{if } D_Z > 0 \text{ and } \theta > \pi/2 \end{cases}$$

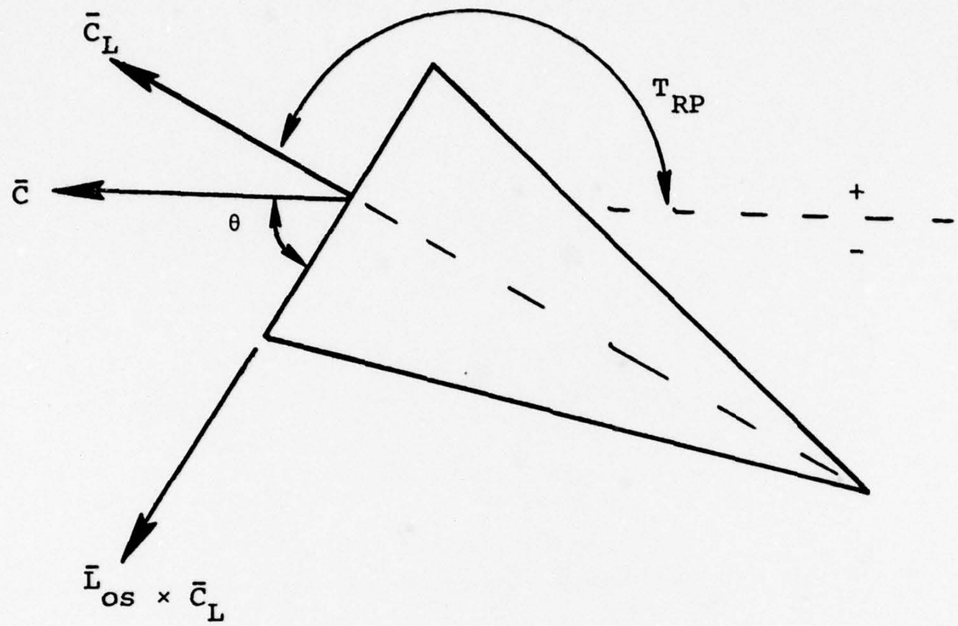


FIGURE 3.4

APPARENT TARGET PLUME AS SEEN FROM THE MISSILE SEEKER
(THE PLANE OF THE PAPER IS PERPENDICULAR TO THE $\bar{L}_{OS}$ VECTOR)

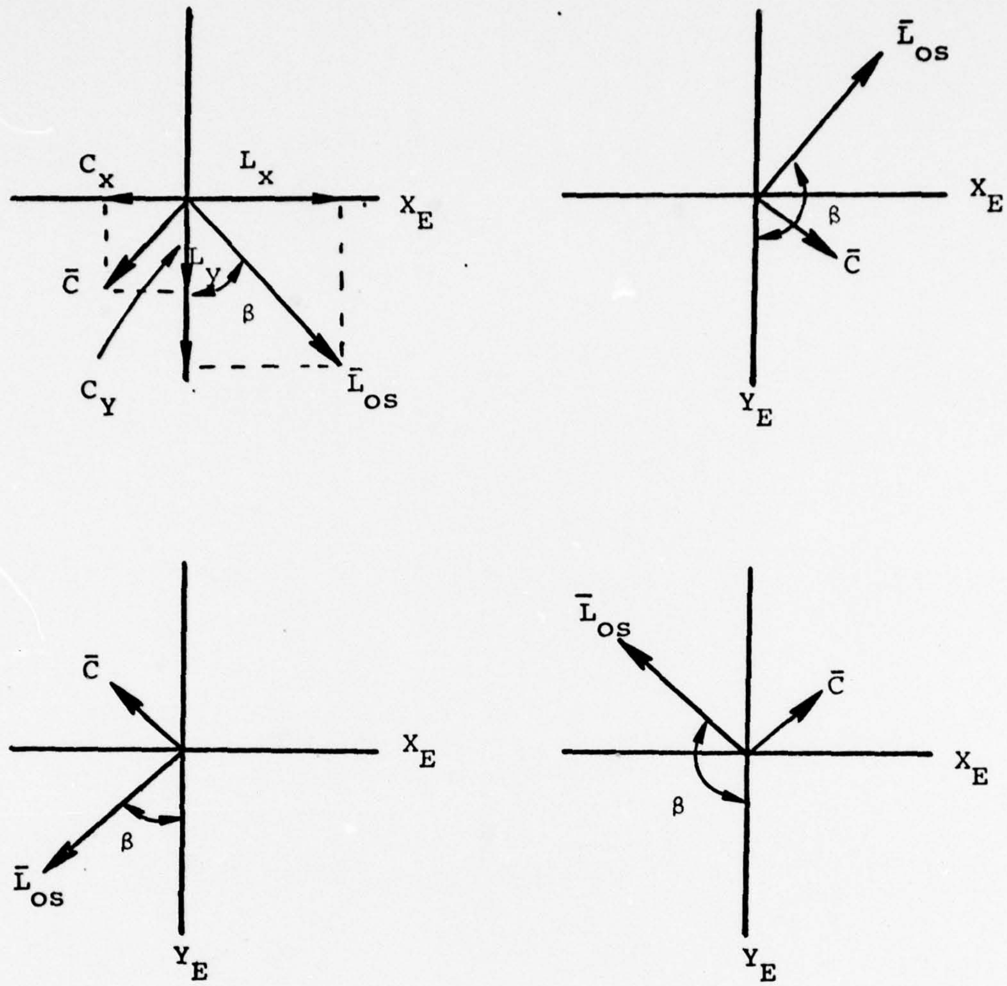


FIGURE 3.5

EXHAUSTION OF GENERAL CONFIGURATIONS FOR $\bar{L}_{OS}$ AND $\bar{C}$

TARGET ANGLE OF ATTACK AND MACH NUMBER

$$\rho = .00237692\ e^{-.00003h_{aSL}} \qquad (\text{SLUGS/FT}^3)$$

$$V_T = \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2 + \dot{Z}_{TE}^2} \qquad (\text{FT/SEC})$$

$$C_{L_{\alpha_T}} = \text{CLAA}\Big|_{Q_M}$$

$$a_s = 1116.89 - .003894\ h_{aSL} \qquad (\text{FT/SEC})$$

$$Q_M = V_T/a_S$$

$$h_{aSL} = \sqrt{X_{ML}^2 + Z_{ML}^2}\ \sin\left[\tan^{-1}\left(\frac{-Z_{ML}}{X_{ML}}\right) + \theta_L\right] + \Delta h^* \qquad (\text{FT})$$

ΔL = Altitude of launch site above sea level (FT)

$$\alpha = \frac{2W_T}{\rho V_T^2\, C_{L_{\alpha_T}}\, g}\left(\sqrt{X_{TE}^2 + Y_{TE}^2 + Z_{TE}^2}\right) \qquad (\text{RAD})$$

*THIS MODEL IS NOT CURRENTLY INCORPORATED IN THE MICOM REAL TIME SIMULATIONS.

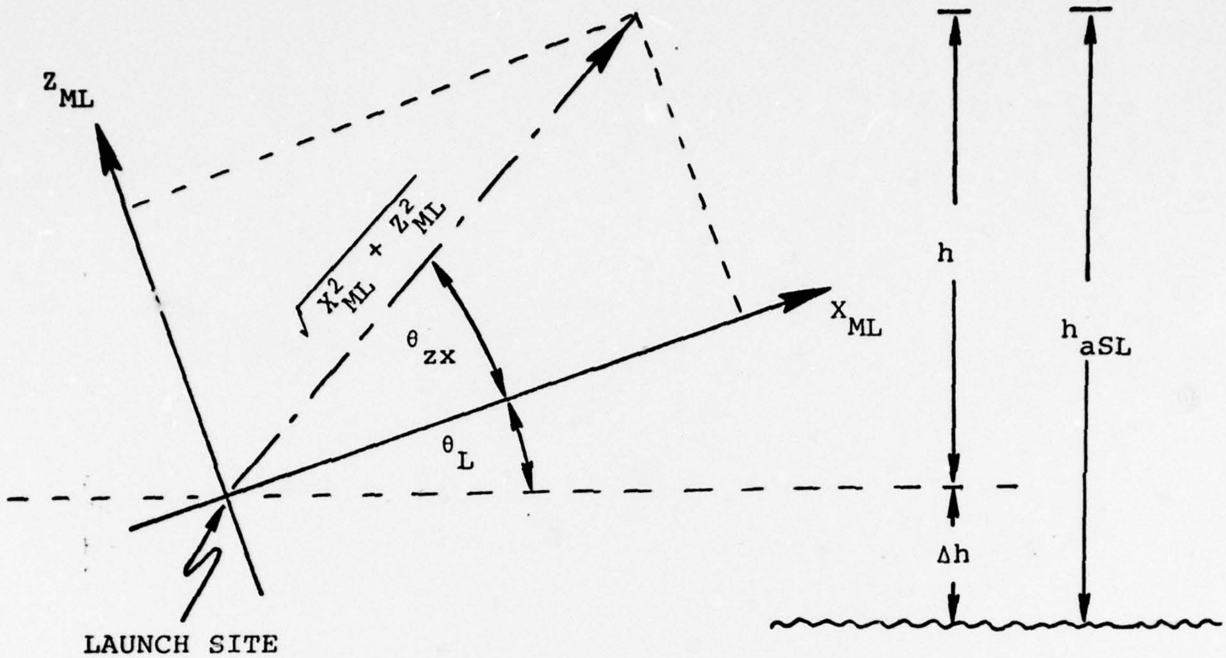


FIGURE 3.6

ALTITUDE OF TARGET ABOVE SEA LEVEL

FLARE TRANSPARENCY RADIUS AND IRIS RATIO

$J_{tu2} = J_{tu2}(t_{FD})$ (W/STER), DFG

$R_F = \sqrt{(X_{ML}-X_{2L})^2 + (Y_{ML}-Y_{2L})^2 + (Z_{ML}-Z_{2L})^2}$

$H_{c2} = J_{tu2} / R_F^2$ (W/CM^2), DFG

$t_2 = t_2 (H_{c2})$ (RAD), DFG

$i_2 = i_2 (H_{c2})$ (RAD), DFG

FLARE AZIMUTH AND ELEVATION

$$\sigma_{Y2} = \frac{Y_{ML}-Y_{2L}}{X_{ML}-X_{2L}} \qquad \text{(RAD)}$$

$$\psi_{24REL} = \begin{cases} 57.3\ (\sigma_{Y4}-\sigma_{Y2})\ ,\ t \geq t_{EJECT} \\ 0,\ t<t_{EJECT} \end{cases} \qquad \text{(DEG.)}$$

$$\dot{\psi}_{24REL} = 57.3 \left\{ \left[\frac{(\dot{Y}_{ML}-\dot{Y}_{4L}) - \sigma_{Y4}\ (\dot{X}_{ML}-\dot{X}_{4L})}{X_{ML}-X_{4L}} \right] - \left[\frac{(\dot{Y}_{ML}-\dot{Y}_{2L}) - \sigma_{Y2}(\dot{X}_{ML}-\dot{X}_{2L})}{X_{ML}-X_{2L}} \right] \right\} \qquad \text{(DEG/SEC)}$$

$$\psi_2 = \psi_4 - \psi_{24REL} \qquad \text{(DEG)}$$

$$\dot{\psi}_2 = \dot{\psi}_4 - \dot{\psi}_{24REL} \qquad \text{(DEG/SEC)}$$

$$\theta_{24REL} = \begin{cases} -57.3(\sigma_{Z4}-\sigma_{Z2}),\ t>t_{EJECT} \\ 0,\ t<t_{EJECT} \end{cases} \qquad \text{(DEG)}$$

$$\dot{\theta}_{24REL} = -57.3 \left\{ \left[\frac{(\dot{Z}_{ML}-\dot{Z}_{4L}) - \sigma_{Z4}(\dot{X}_{ML}-\dot{X}_{4L})}{(X_{ML}-X_{4L})} \right] - \left[\frac{(\dot{Z}_{ML}-\dot{Z}_{2L}) - \sigma_{Z2}(\dot{X}_{ML}-\dot{X}_{2L})}{(X_{ML}-X_{2L})} \right] \right\} \qquad \text{(DEG/SEC)}$$

FLARE DYNAMICS

$$X_{2L} = \frac{-\rho\ C_{D2}\ S_2 V_2\ X_{2L}}{2M_2} - g \sin\theta_L\ ,\ t>t_{EJECT}$$

$$Y_{2L} = \frac{-\rho\ C_{D2} S_2\ V_2\ X_{2L}}{2M_2}\ ,\ t>t_{EJECT}$$

$$Z_{2L} = \frac{-\rho\ C_{D2} S_2\ V_2\ Z_{2L}}{2M_2} + g \cos\theta_L\ ,\ t>t \qquad (FT/SEC^2)$$

$$\left.\begin{aligned} X_{2L} &= X_{4L} \\ Y_{2L} &= Y_{4L} \\ Z_{2L} &= Z_{4L} \\ \dot{X}_{2L} &= \dot{X}_{4L} \\ \dot{Y}_{2L} &= \dot{Y}_{4L} \\ \dot{Z}_{2L} &= \dot{Z}_{4L} \end{aligned}\right\}\ t<t_{EJECT} \qquad (FT)$$

TAILPIPE AZIMUTH AND ELEVATION

$$\sigma_{Y4} = \frac{Y_{ML}-Y_{4L}}{X_{ML}-X_{4L}} \qquad \text{(RAD)}$$

$$\psi_4 = 57.3\ \sigma_{Y4} \qquad \text{(DEG)}$$

$$\dot{\psi}_4 = 57.3 \left[\frac{(\dot{Y}_{ML}-\dot{Y}_{4L}) - \sigma_{Y4}(\dot{X}_{ML}-\dot{X}_{4L})}{(X_{ML}-X_{4L})} \right] \qquad \text{(DEG/SEC)}$$

$$\sigma_{Z4} = \frac{Z_{ML}-Z_{4L}}{X_{ML}-X_{4L}} \qquad \text{(RAD)}$$

$$\theta_4 = -57.3\ \sigma_{Z4} \qquad \text{(DEG)}$$

$$\dot{\theta}_4 = -57.3 \left[\frac{(\dot{Z}_{ML}-\dot{Z}_{4L}) - \sigma_{Z4}(\dot{X}_{ML}-\dot{X}_{4L})}{(X_{ML}-X_{4L})} \right] \qquad \text{(DEG/SEC)}$$

TRANSFORMATION OF $\bar{C}_L$ AND $\bar{L}_{os}$ FROM LAUNCH TO EARTH COORDINATES

$$\bar{C}_L = \bar{i}_E \left\{ \frac{X_{TE}}{V_T} \cos(\alpha) + \frac{X_{TE}\ Z_{TE}\ \mathrm{SIN}(\alpha)}{V_T \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}} \right\}$$

$$+ \bar{j}_E \left\{ \frac{Y_{TE}}{V_T} \cos(\alpha) + \frac{Y_{TE}\ Z_{TE}\ \sin(\alpha)}{V_T \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}} \right\}$$

$$+ \bar{K}_E \left\{ \frac{Z_{TE} \cos(\alpha)}{V_T} - \frac{\sqrt{X_{TE}^2 + Y_{TE}^2}\ \sin(\alpha)}{V_T} \right\}$$

$$\bar{L}_{os} = \bar{i}_E\ (X_G \cos\theta_L \cos\psi_L - Y_G \sin\psi_L + Z_G \sin\theta_L \cos\psi_L)$$

$$+ \bar{j}_E\ (X_G \cos\theta_L \sin\psi_L + Y_G \cos\psi_L + Z_G \sin\theta_L \sin\psi_L)$$

$$+ \bar{K}_E\ (-X_G \sin\theta_L + Z_G \cos\theta_L)$$

$$X_G = X_{ML} - X_{TL}$$

$$Y_G = Y_{ML} - Y_{TL}$$

$$Z_G = Z_{ML} - Z_{TL}$$

VARIABLE SCALE FACTOR

$$K = \begin{cases} 1 + \dfrac{\gamma t}{t'}, & t<t' \\ 1 + \gamma & t>t' \end{cases}$$

TABLE 3.1
DESCRIPTION OF PROGRAM SYMBOLS

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
________	SPO	Scaled plume rotation angle
T_{RP}	TRP	Plume rotation angle (RAD.)
θ_L	THETAL	Initial elevation angle of target (DEG.)
t	DT	Real time (SEC.)
g	________	Gravitational constant
x_E	XE	Target aerodynamic acceleration table in earth fixed coordinates (Note ZE includes an added factor of 32.174 FT/SEC/SEC
y_E	YE	
z_E	ZE	
α	Al	Target angle of attack (RAD.)
t'	G,GAM	Time at approximately 1000 feet to go
C_{LA}	CLA	Aerodynamic lift coefficient due to angle of attack
$\bar{V}_{TI}$	VTI	Target inertial velocity
ℓ/B	RLB	Apparent plume length to breadth ratio
E	________	Subscript which denotes earth fixed coordinates

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
GN	________	Subscript which denotes generalized target coordinates
F	________	Subscript which denotes target fixed coordinates
G	________	Subscript which denotes guidance coordinates
L	________	Subscript which denotes launch coordinates
R_i	RI	Initial range (FT.)
R	RFEET	Range
________	EDOT	Scaled γ/t'
ρ	RHO	Air density (SLUGS/FT^3)
γ_{min}	GGG	Minima of the overload function
Scalet	SCALET	Scale Factor = 10.2375
K	SKK	Variable scale factor, $K=\begin{cases} 1+\frac{\gamma t}{t'}, & t\leq t' \\ 1+\gamma, & t>t' \end{cases}$
________	XX	Range table for data collection
________	XXS	Scaled range table (XX)

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
I_{r7}	RN	For MICOM Hybrid, a uniform random number, RN ε (-1,1) For IRSS, iris ratio number 7
Cos (ε)	COSE	Cosine of angle between LOS and center line of target
$x_{ML} - x_{TL}$	DX	Scaled xxx, yyy, zzz
$y_{ML} - y_{TL}$	DY	
$z_{ML} - z_{TL}$	DZ	
t	DT	Real time
$x_{ML} - x_{TL}$	XXX	x-Missile minus x-target position (FT.)
$y_{ML} - y_{TL}$	YYY	y-Missile minus y-target position (FT.)
$z_{ML} - z_{TL}$	ZZZ	z-Missile minus z-target position (FT.)
$\dot{x}_{ML} - \dot{x}_{TL}$	XDOT	x-Missile minus x-target velocity (FT./SEC.)
$\dot{y}_{ML} - \dot{y}_{TL}$	YDOT	y-Missile minus y-target velocity (FT./SEC.)
$\dot{z}_{ML} - \dot{z}_{TL}$	ZDOT	z-Missile minus z-target velocity (FT./SEC.)

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
$\dot{x}_{TL}$	XDTGMS	Tables of target velocity components in launch system
$\dot{y}_{TL}$	YDTGMS	
$\dot{z}_{TL}$	ZDTGMS	
ℓ	———	Apparent plume length (FT.)
B	———	Apparent plume breadth (FT.)
β	———	$\cos^{-1} (L_y / \sqrt{L_x^2+L_y^2})$
h	ZALT	Altitude of missile above sea level (FT.)
———	XCOMP	Interpolated value of XDTGMS
———	YCOMP	Interpolated value of YDTGMS
———	ZCOMP	Interpolated value of ZDTGMS
$\dot{x}_{TE}$	XC	Interpolated value of XDM
$\dot{y}_{TE}$	YC	Interpolated value of YDM
$\dot{z}_{TE}$	ZC	Interpolated value of ZDM

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
———	XDM	Tables of target velocity components in earth system
———	YDM	
———	ZDM	
———	TIME	
———	PPX	
———	PPY	Storage for missile position, velocity and time in the region of pre-specified range table entries
———	PPZ	
———	VMX	
———	VMY	
———	VMZ	
———	IPTS	Number of points in range table
———	MISSED	An array of values representing miss conditions
———	T111	Angle between $(\overline{LOS} \times \overline{C}_L)$ and horizon reference vector
———	DAC1,DAC2, DAC3,..., DAC11	DACS transmitted from CDC-6600 to AD/4

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
x_G	———	x-Missile minus x-target (in launch coordinates)
y_G	———	y-Missile minus y-target (in launch coordinates)
z_G	———	z-Missile minus z-target (in launch coordinates)
VT	VTI	Target inertial velocity
———	XTA	Interpolated target accelerations in earth fixed coordinate system
———	YTA	
———	ZTA	
$\overline{C}_L$	———	Longitudinal center line of target
C_{Lx}	G1	Components of $\overline{C}_L$ in earth fixed coordinates
C_{Ly}	G2	
C_{Lz}	G3	
$\overline{L}_{os}$	———	Line of sight vector

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
L_x	F1	Components of $\bar{L}_{os}$ in earth fixed coordinates
L_y	F2	
L_z	F3	
x_{ML}	———	x-Missile position in launch coordinates
y_{ML}	———	y-Missile position in launch coordinates
z_{ML}	———	z-Missile position in launch coordinates
x_{TL}	———	x-target position in launch coordinates
y_{TL}	———	y-target position in launch coordinates
z_{TL}	———	z-target position in launch coordinates
x_{TE}	———	x-position of target in inertial coordinates
y_{TE}	———	y-position of target in inertial coordinates
z_{TE}	———	z-position of target in inertial coordinates

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
———	XDTGO	Initial values of XDTGMS, YDTGMS, ZDTGMS
———	YDTGO	
———	ZDTGO	
H_{c7}	———	Commanded irradiance for plume
f	19.5	Focal length of projection lens (CM)
$H\Big\vert_{\substack{R=7000\\ \varepsilon}}$	HR7E	Irradiance at 7000 meters (W/CM^2)
P_1	P1IRSS	Vertex angle of plume transparency (DEG)
P_2	P2IRSS	Length of plume transparency (IN.)
A_{t7}	AT7	Area of plume transparency (IN^2)
$J_{tu1,plume}$	PJTU1	Available radiant intensity of target as function of plume transparency area taken from calibration in IRSS (W/STER)
$T_{f,plume}$	0.3	Neutral density transmission factor for plume

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
	DXG DYG DZG XDO YDO ZDO	Dummy storage location for current value of DX, DY, DZ, XDOT, YDOT and ZDOT respectively. These variables are returned to the main program for diagnostic purposes
	ADC1,ADC2, ADC3,..., ADC10	ADCs transmitted from AD/4 to CDC-6600
	LEVEL	Status of maneuver -7 implies not in real time 0 implies in real time +7 implies target traj table exceeded
	WMAN	Flag, =7 implies collect data, =0 implies don't collect data
	MAN	Array for storing target velocities at each range table entry.
	INDEX	Index of range table entries
	KCK	Flag, =1 implies $t>t'$, = -1 implies $t \le t'$
	XMISS	Array to save ADCs under miss condition (See MISSED)

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
W_T	WT	Weight of target
S_T	ST	Reference surface area of target
$\bar{C}$	________	Horizon reference vector
Q_M	QM	Mach number of target
a_s	AAA	Speed of sound
V_2	________	Speed of flare
R_F	________	Range to go between missile and flare
H_{i2}, H_{c2}	________	Irradiance of flare W/CM^2 (Note, letter c denotes calibration)
$\dot{X}_{FL}$	________	Velocity of flare in launch coordinates (FT/SEC)
$\dot{Y}_{FL}$	________	
$\dot{Z}_{FL}$	________	
σ_{Y4}	________	Approximate ψ_4 (RAD)
σ_{Z4}	________	Approximate θ_4 (RAD)
σ_{Y2}	σ_{YT}	Approximate ψ_2 (RAD)

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
σ_{Z2}	σ_{ZT}	Approximate θ_2 (RAD)
________	J_{tu2}	Available radiant intensity of flare
________	T_{FD}	Time of flare drop (SEG)
$\|L_{os}\|$	R	Norm of line of sight (FT)
t_{FD}	t_{FD}, t_{fd}	Time associated with flare drop
δ	________	Missile wing deflection
ψ_2	ψ_{F2}	Flare azimuth (DEG)
ψ_4	ψ_{T4}	Tail pipe azimuth (DEG)
ψ_7	ψ_{T7}	Plume azimuth (DEG)
ψ_{24}	ψ_{T2REL}	
θ_2	θ_{T2}	Flare elevation (DEG)
θ_4		Tail pipe elevation (DEG)
θ_7	θ_{T7}	Plume elevation (DEG)
θ_{24}		
$\dot{\psi}_2$	$\dot{\psi}_{F2}$	Flare azimuth rate (DEG/SEC)
$\dot{\psi}_4$	$\dot{\psi}_{T4}$	Tail pipe azimuth rate (DEG/SEC)

VARIABLE	PROGRAM SYMBOL	DESCRIPTION	
$\dot{\psi}_7$	$\dot{\psi}_{T7}$	Plume azimuth rate (DEG/SEC)	
$\dot{\psi}_{24}$	$\dot{\psi}_{T2REL}$		
$\dot{\theta}_2$	$\dot{\theta}_{T2}$	Flare elevation rate (DEG/SEC)	
$\dot{\theta}_4$	$\dot{\theta}_{T4}$	Tail pipe elevation rate (DEG/SEC)	
$\dot{\theta}_7$	$\dot{\theta}_{T7}$	Plume elevation rate (DEG/SEC)	
$\dot{\theta}_{24}$	$\dot{\theta}_{T2REL}$		
t_2	———	Transparency radius for flare	used to command true size
t_4	———	Transparency radius for tailpipe	
P	θ	GUM pitch angle (DEG)	
Y	ψ	GUM yaw angle (DEG)	
R	ϕ	GUM roll angle (DEG)	
r'	———	GUM Yaw rate (DEG/SEC)	
q'	———	GUM pitch rate (DEG/SEC)	
p'	———	GUM roll rate (DEG/SEC)	

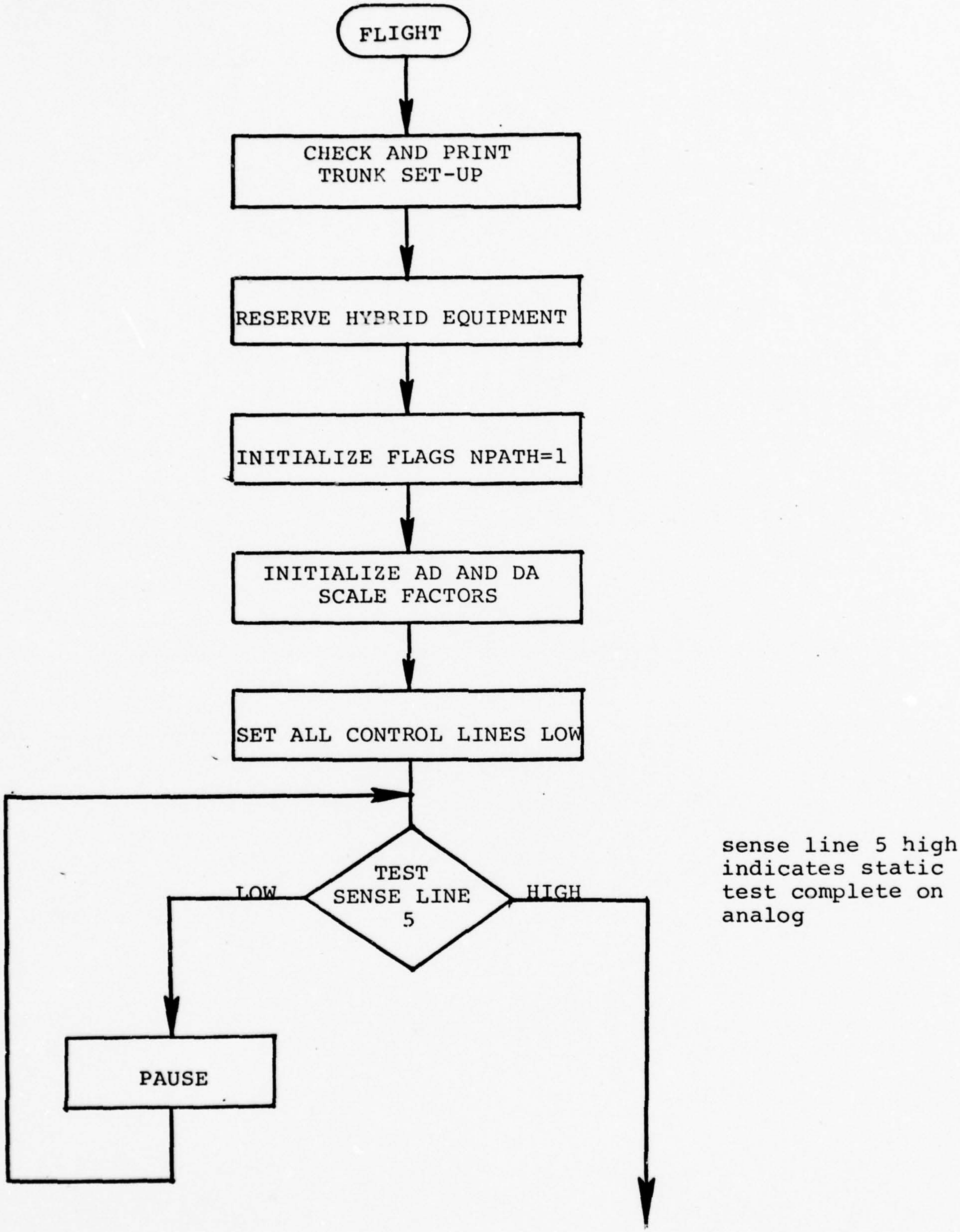
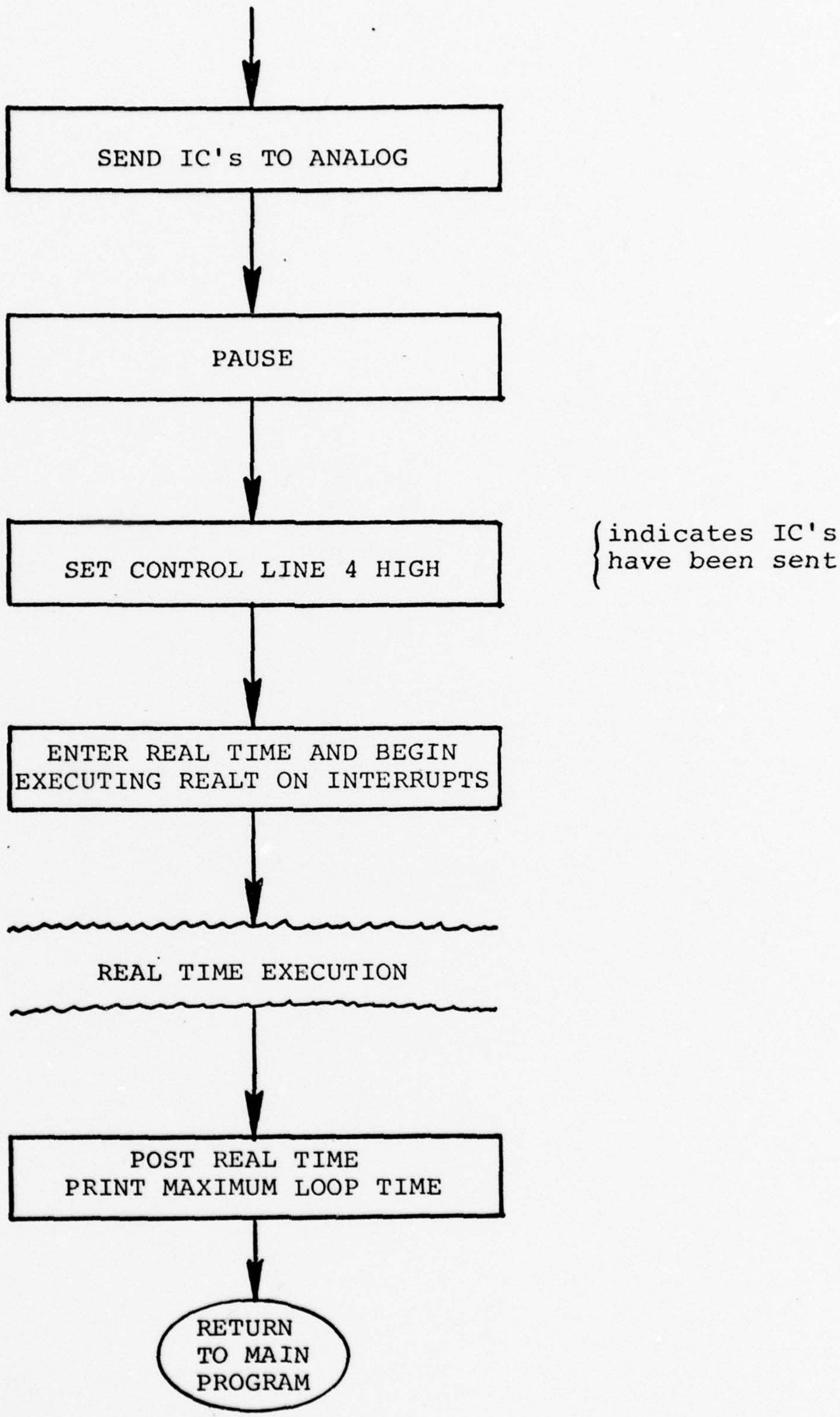
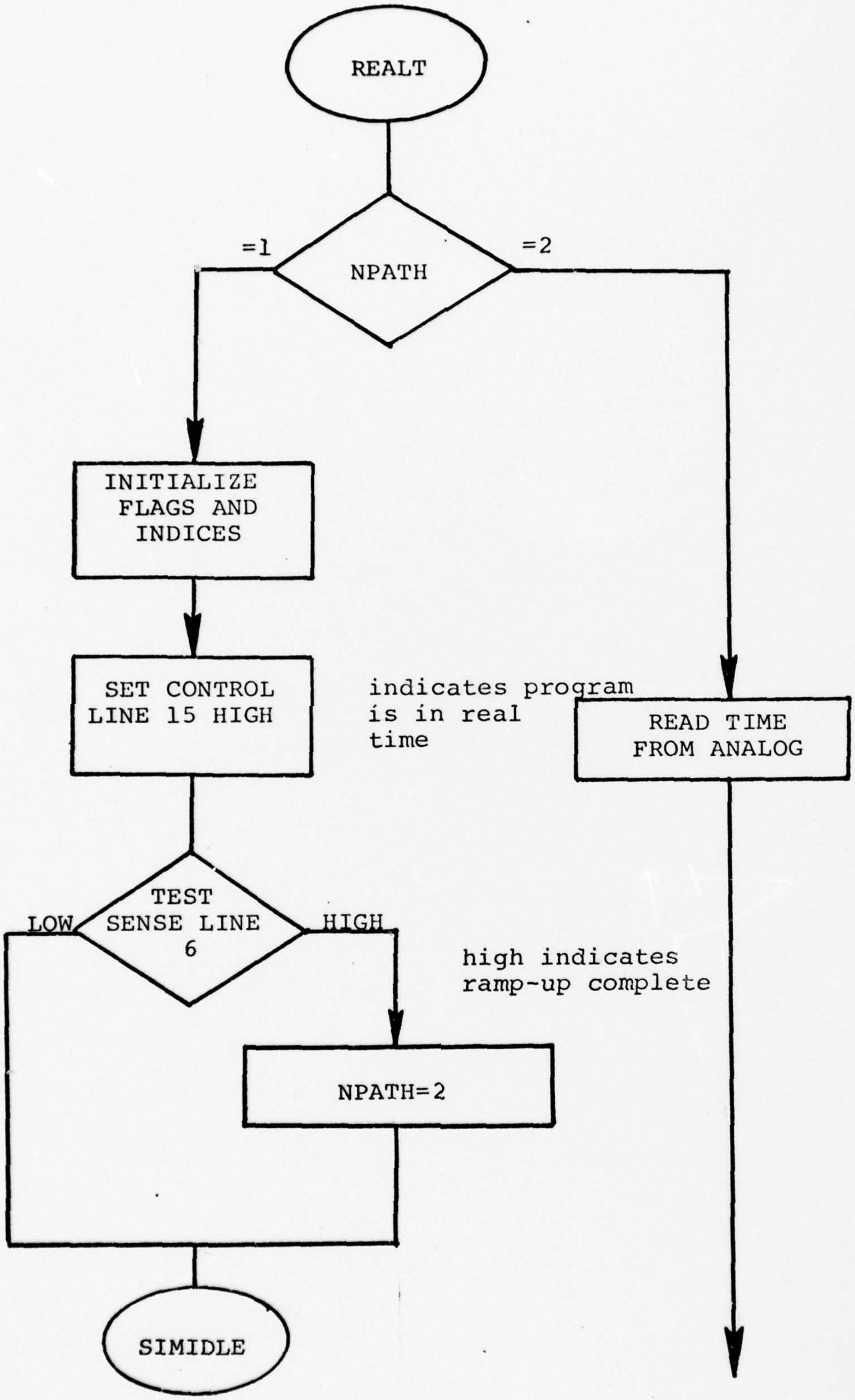
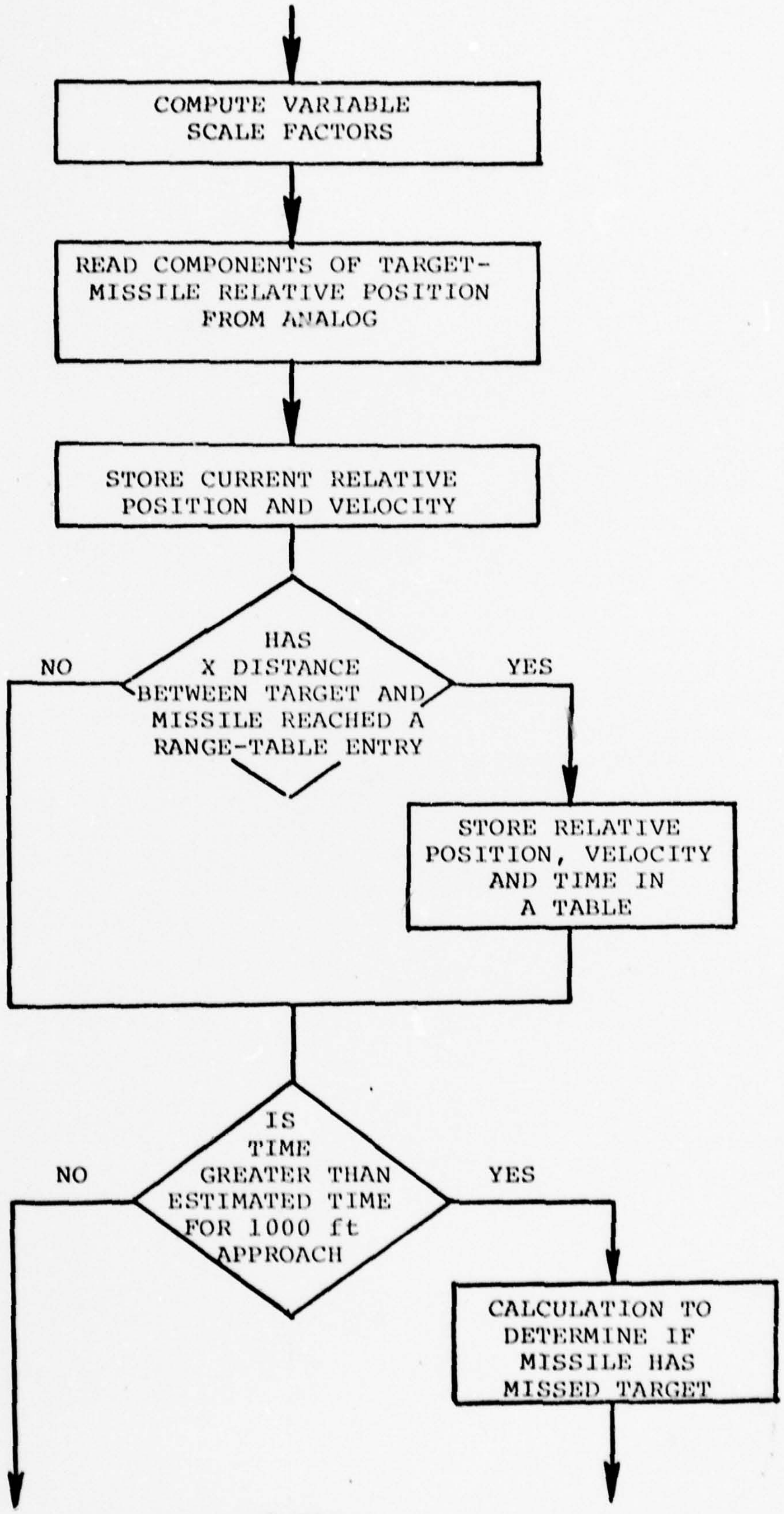


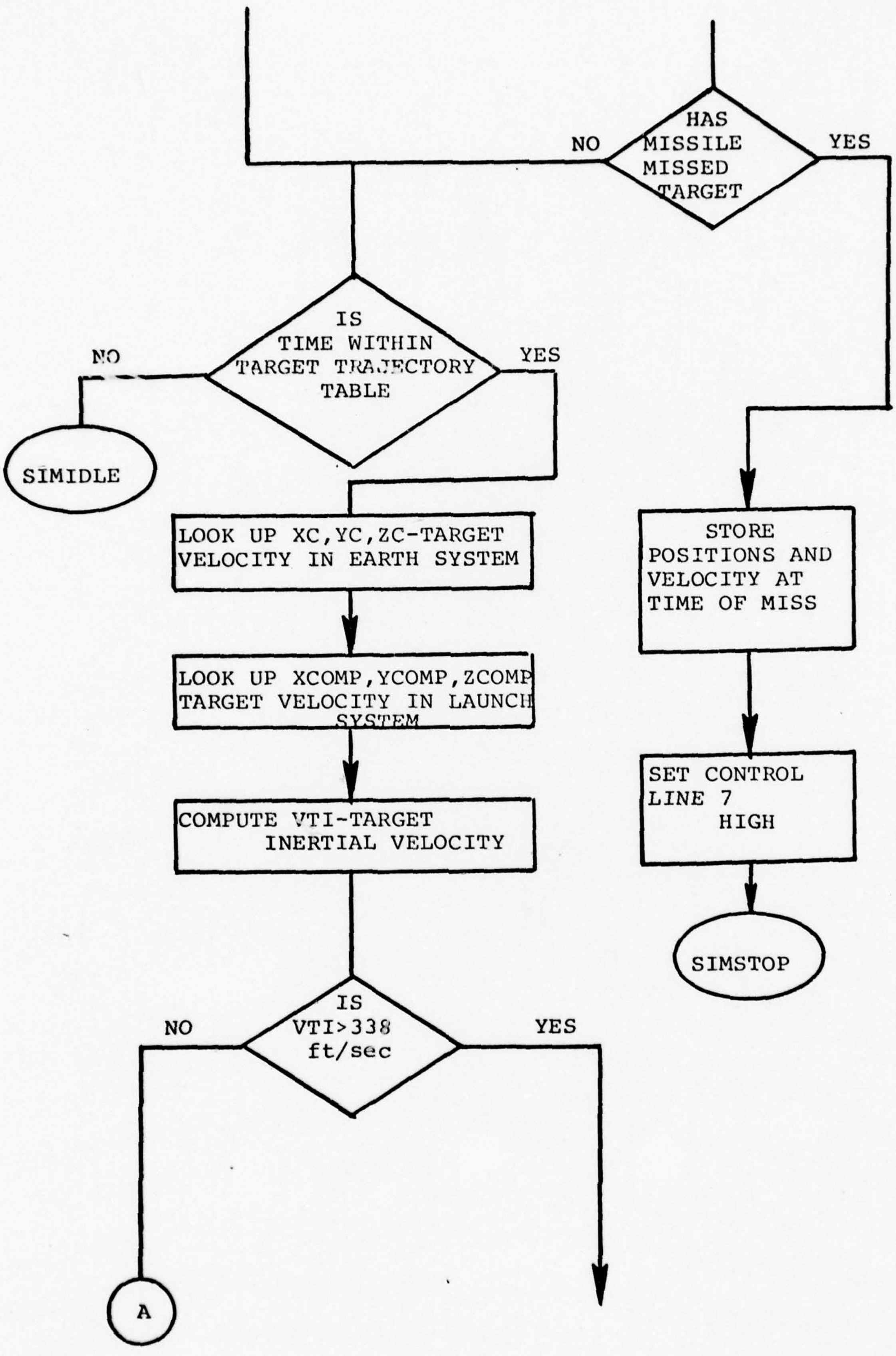
FIGURE 3.7
FLOW CHART FOR SUBROUTINE FLIGHT

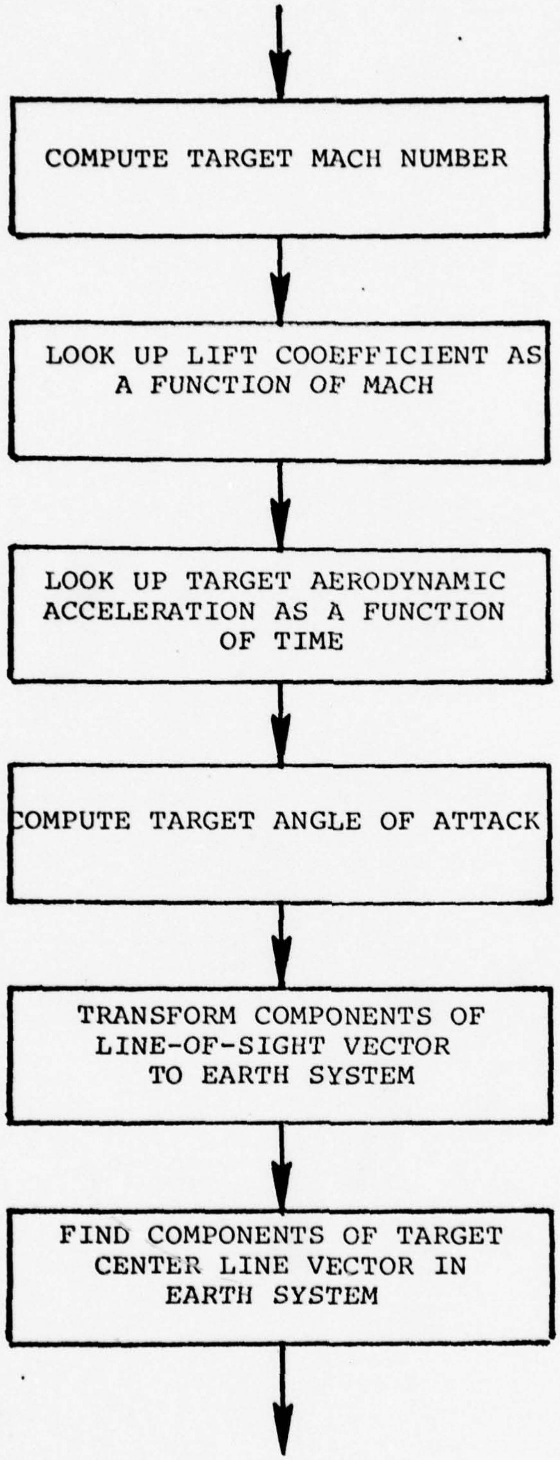

SEND IC's TO ANALOG
PAUSE
SET CONTROL LINE 4 HIGH
indicates IC's have been sent
ENTER REAL TIME AND BEGIN EXECUTING REALT ON INTERRUPTS
REAL TIME EXECUTION
POST REAL TIME PRINT MAXIMUM LOOP TIME
RETURN TO MAIN PROGRAM

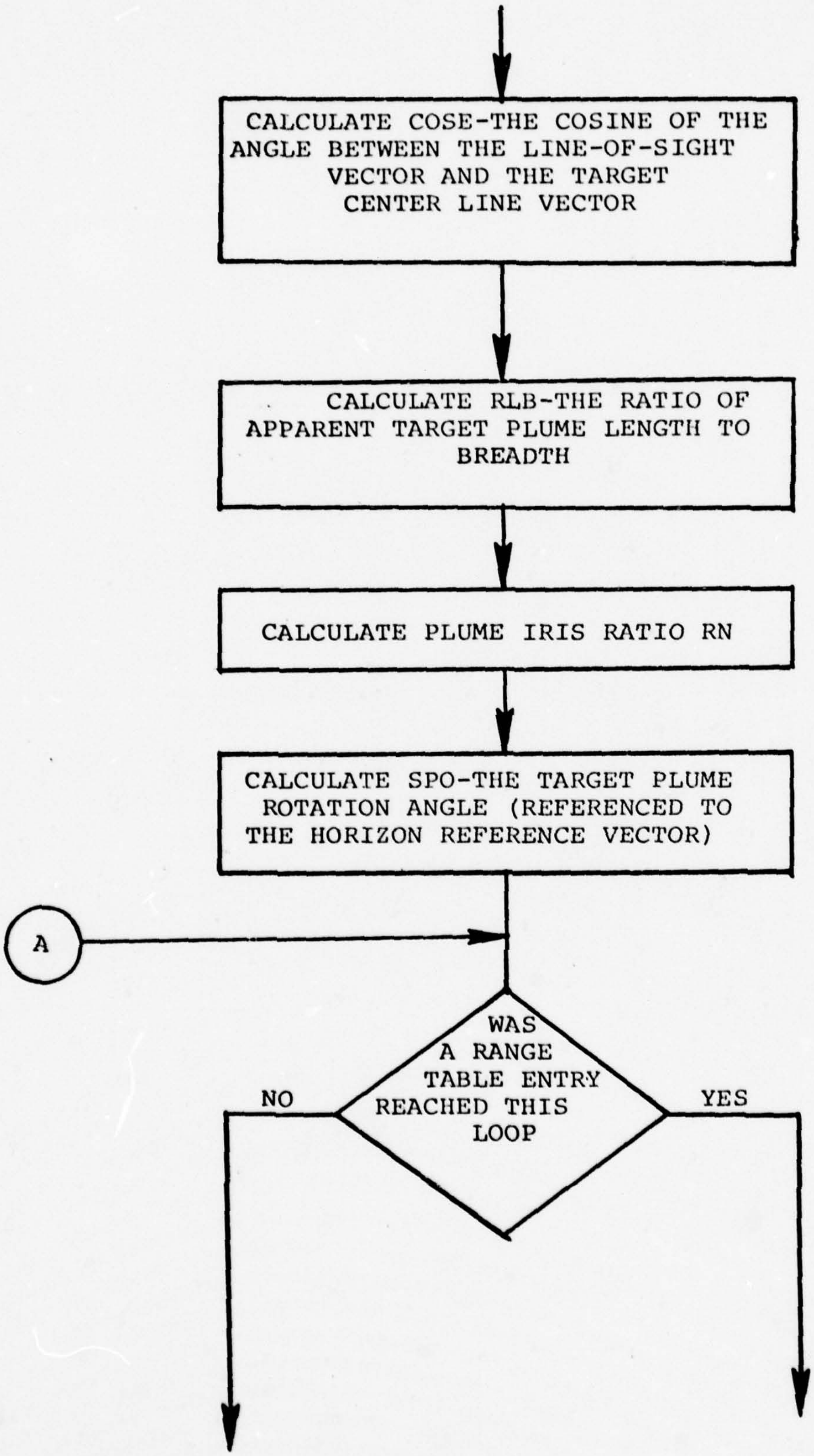


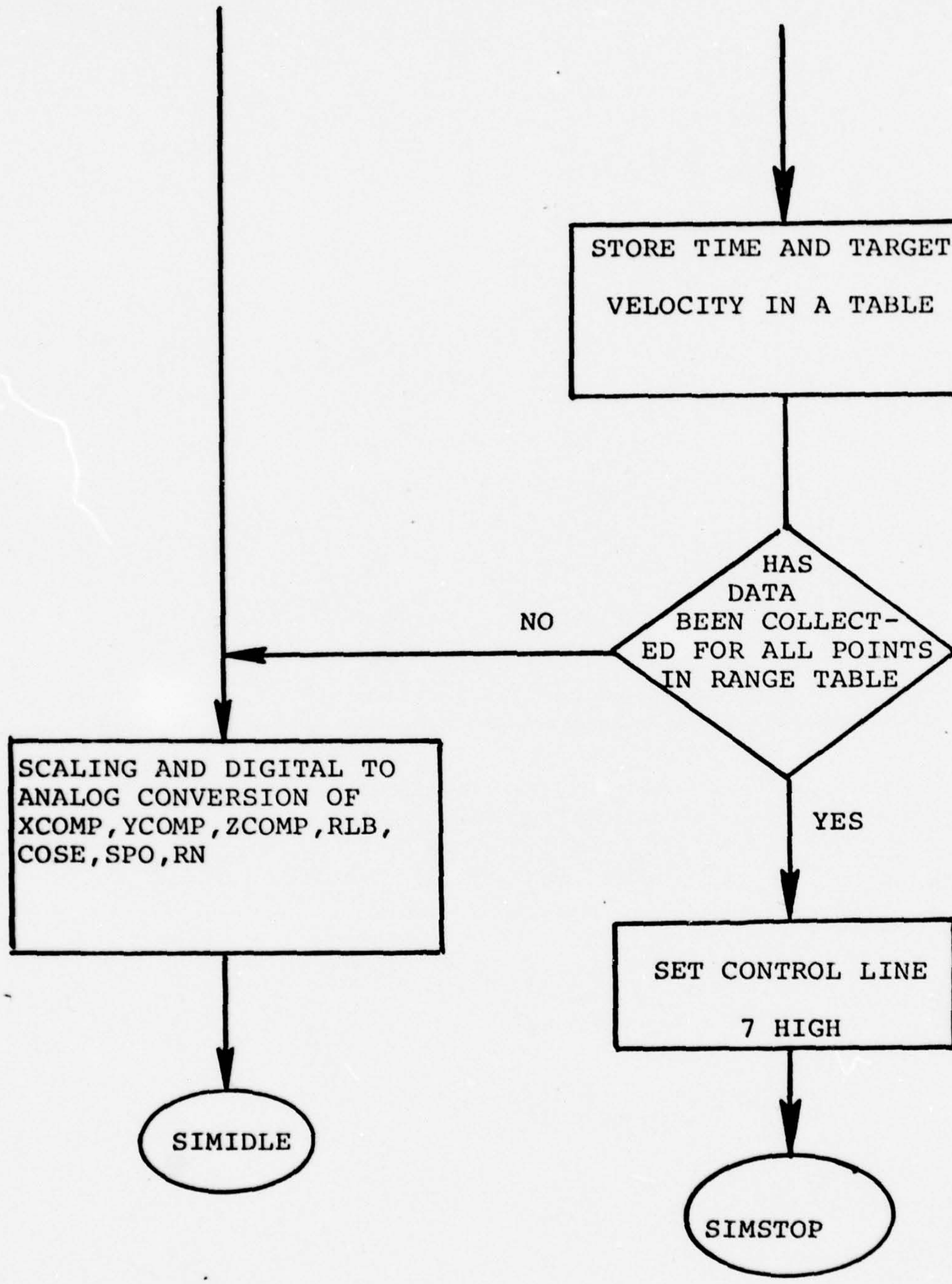
FLOW CHART FOR SUBROUTINE REALT


COMPUTE VARIABLE SCALE FACTORS
READ COMPONENTS OF TARGET-MISSILE RELATIVE POSITION FROM ANALOG
STORE CURRENT RELATIVE POSITION AND VELOCITY
HAS X DISTANCE BETWEEN TARGET AND MISSILE REACHED A RANGE-TABLE ENTRY
NO
YES
STORE RELATIVE POSITION, VELOCITY AND TIME IN A TABLE
IS TIME GREATER THAN ESTIMATED TIME FOR 1000 ft APPROACH
NO
YES
CALCULATION TO DETERMINE IF MISSILE HAS MISSED TARGET


HAS MISSILE MISSED TARGET
NO
YES
IS TIME WITHIN TARGET TRAJECTORY TABLE
NO
YES
SIMIDLE
LOOK UP XC,YC,ZC-TARGET VELOCITY IN EARTH SYSTEM
LOOK UP XCOMP,YCOMP,ZCOMP TARGET VELOCITY IN LAUNCH SYSTEM
COMPUTE VTI-TARGET INERTIAL VELOCITY
IS VTI>338 ft/sec
NO
YES
A
STORE POSITIONS AND VELOCITY AT TIME OF MISS
SET CONTROL LINE 7 HIGH
SIMSTOP


COMPUTE TARGET MACH NUMBER
LOOK UP LIFT COOEFFICIENT AS A FUNCTION OF MACH
LOOK UP TARGET AERODYNAMIC ACCELERATION AS A FUNCTION OF TIME
COMPUTE TARGET ANGLE OF ATTACK
TRANSFORM COMPONENTS OF LINE-OF-SIGHT VECTOR TO EARTH SYSTEM
FIND COMPONENTS OF TARGET CENTER LINE VECTOR IN EARTH SYSTEM


CALCULATE COSE-THE COSINE OF THE ANGLE BETWEEN THE LINE-OF-SIGHT VECTOR AND THE TARGET CENTER LINE VECTOR
CALCULATE RLB-THE RATIO OF APPARENT TARGET PLUME LENGTH TO BREADTH
CALCULATE PLUME IRIS RATIO RN
CALCULATE SPO-THE TARGET PLUME ROTATION ANGLE (REFERENCED TO THE HORIZON REFERENCE VECTOR)
A
WAS A RANGE TABLE ENTRY REACHED THIS LOOP
NO
YES


STORE TIME AND TARGET VELOCITY IN A TABLE
HAS DATA BEEN COLLECT-ED FOR ALL POINTS IN RANGE TABLE
NO
YES
SCALING AND DIGITAL TO ANALOG CONVERSION OF XCOMP,YCOMP,ZCOMP,RLB, COSE,SPO,RN
SET CONTROL LINE 7 HIGH
SIMIDLE
SIMSTOP

3.2 STINGER REAL TIME/INPUT-OUTPUT REQUIREMENTS

A description of the real time I/O requirements for the STINGER/IRSS simulation is given in this section. These I/O requirements are quite complex and involve both analog and digital information. In Figure 3.8 an overview of I/O for the STINGER/IRSS simulation is given. This configuration involves the interconnection of digital computers, analog computers and real time hardware.

In Tables 3.2 through 3.10 the input/output signals are identified with respect to purpose/physical description and current (or typical) usage. In each case these tables should be read with the aid of information presented in section 3.1.

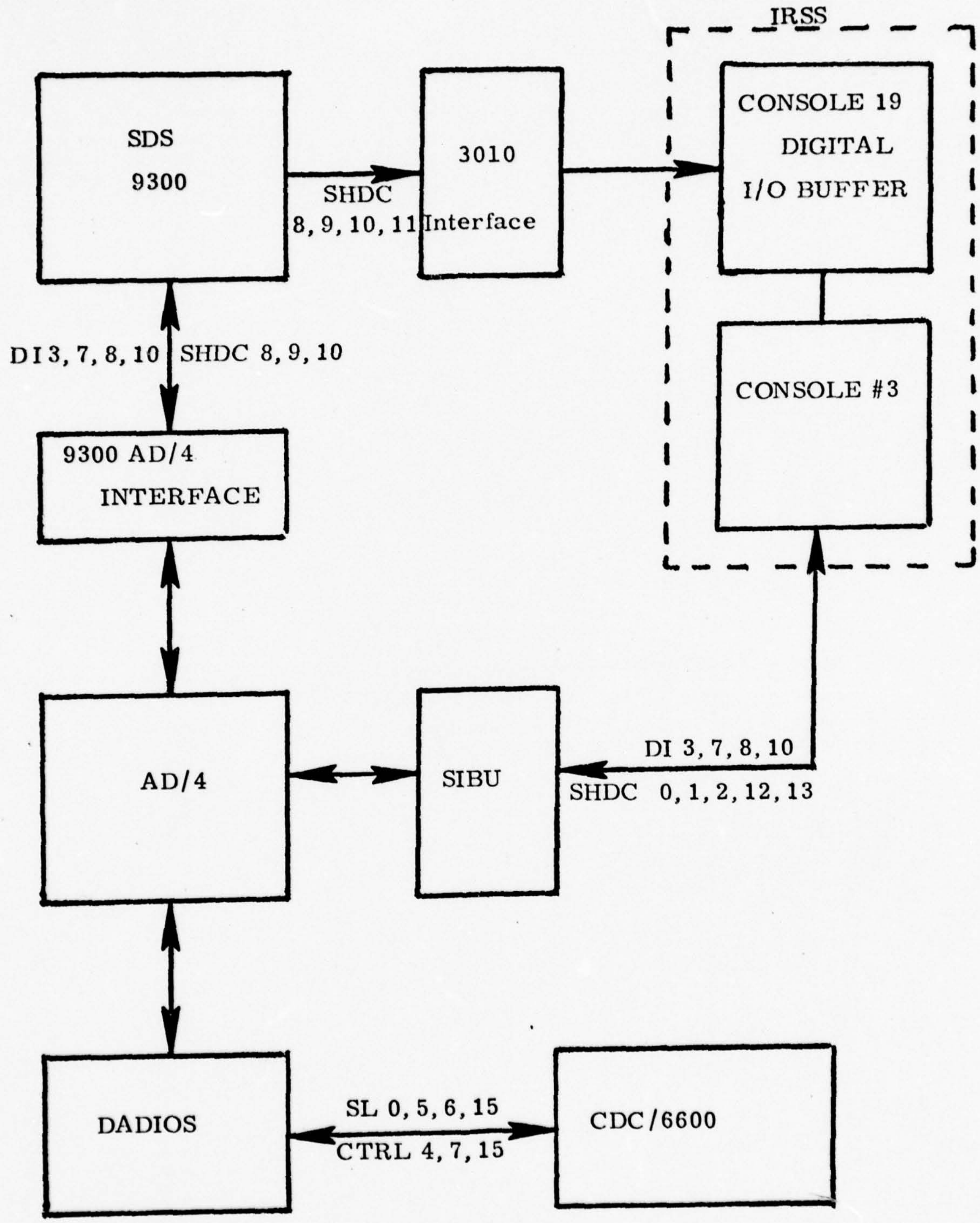


FIGURE 3.8

OVERVIEW OF I/O FOR STINGER REAL TIME SIMULATION

TABLE 3.2
CONTROL LINE ASSIGNMENT
CDC OUTPUT DISCRETES

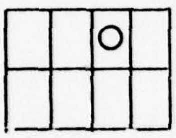
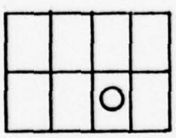
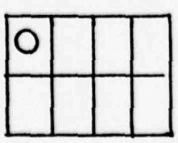
CDC-6600 VARIABLE	CDC-6600 BIT	AD/4#2 TRUNK LINE	PURPOSE/ACTION
CTRL0-CTRL3	0 - 3	TR37-TR34	Not in use
CTRL4 	4	TR33	Is set high after ICs have been sent to analog. It is also the signal that starts the analog (SYS-OP) for MICOM hybrid
CTRL5, CTRL6	5,6	TR32, TR31	Not in use
CTRL7 	7	TR30	Is set high when the simulation is terminated and the analog should go into system hold. Termination can occur for "missed target" or "all data collected"
CTRL8-CTRL14	8 - 14	TR17-TR11	Not in use
CTRL15 	15	TR10	Real time looping CDC/6600 waiting for SL6 HIGH

TABLE 3.3
SENSE LINE ASSIGNMENT
(CDC INPUT DISCRETES)

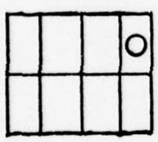
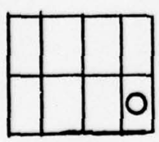
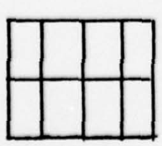
CDC-6600 VARIABLE	CDC-6600 BIT	AD/4#2 TRUNK LINE	PURPOSE/ACTION
SL0	0	TR27	Post real time control flag, HIGH branches to post real time LO branches to ICs
SL1-SL3	1 - 3	TR26-TR24	Not in use
SL4	4	TR23	Not in use
SL5 	5	TR22	Set high by the static check OK switch on AD/4. This must be high to send ICs to analog
SL6 	6	TR21	Set high by the static check OK switch on the AD/4 for MICOM Hybrid, or set high by the ramp up circuit for IRSS hybrid when initial IRSS conditions are attained
SL7-SL14	7 - 14	TR20 and TR07-TR01	Not in use
SL15 	15	TR00	Abnormal abort for CDC/6600 if set HIGH (AD/4 manual switch)

TABLE 3.4
SHUTTER AND DISCRETE COMMANDS (SHDC)

SHDC BIT	AD/4 TRUNK	DESCRIPTION
0	10	Open shutter #2, generated by manual switch at AD/4 console
1	12	Open shutter #4, generated by manual switch at AD/4 console
2	14	Open shutter #7, generated by manual switch at AD/4 console
8	04	Simulation running, generated by XDS-9300
9	16	Field transmitted, generated by XDS-9300 in response to DI10
10	06	End of problem (normal termination in closed loop), generated by AD/4
11	03	Start simulation, generated by XDS-9300
12	02	Target acquired, generated by manual switch at AD/4 console
13	00	Discrete 5, switch seeker from external to internal power (generated by AD/4)

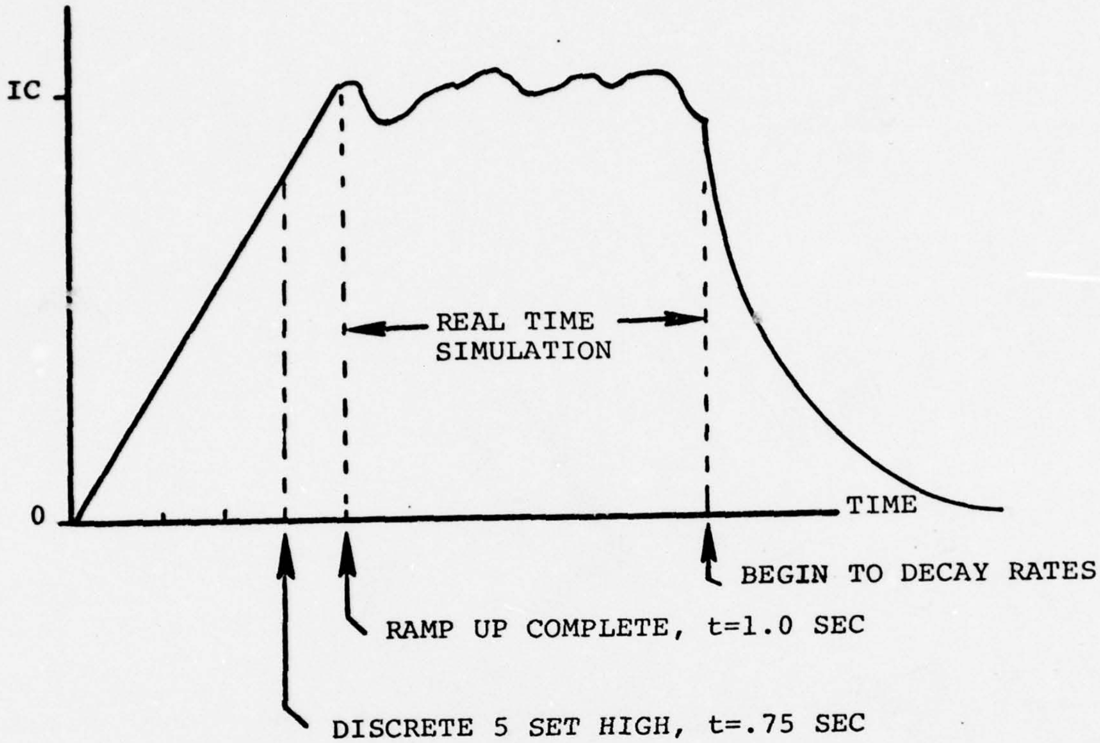


FIGURE 3.9

RATE RAMP UP AND DECAY FOR REAL TIME SIMULATIONS

Field transmitted is a signal to the GEPAC during closed loop operation that the first frame of data is available. This occurs after the CDC-6600 has received an initialization request (DI10). Discrete 5 is set high approximately .25 seconds prior to ramp up completed. This switches the seeker power from external to internal power to simulate an actual launch.

TABLE 3.5
DIGITAL INPUT DISCRETES (DI)

DI BIT	AD/4 TRUNK	DESCRIPTION
3	T11	IRSS ready, sent to XDS-9300 via AD/4
7	T07	End of problem
8	T05	Emergency Shut Down
10	T01	Initial field request

TABLE 3.6

AD/4-CDC/6600 ADC/ASSIGNMENTS

ANALOG VARIABLE NAME	AD/4 ANLG. VOLTAGE $-100<V<+100$	DIGITAL FRACTION $-1<N<+1$	CDC/6600 ADC ASSIGN.	AD/4 TRUNK LINE ASSIGNMENT
DX	$\frac{K(X_{ML}-X_{TL})}{20}$	$\frac{K(X_{ML}-X_{TL})}{(20)(100)}$	2	151
DY	$\frac{K(Y_{ML}-Y_{TL})}{20}$	$\frac{K(Y_{ML}-Y_{TL})}{(20)(100)}$	3	152
DZ	$\frac{K(Z_{ML}-Z_{TL})}{20}$	$\frac{K(Z_{ML}-Z_{TL})}{(20)(100)}$	4	153
DT	$10t$	$\frac{10t}{(100)}$	5	154
XDOT	$\frac{\dot{X}_{ML}-\dot{X}_{TL}}{200}$	$\frac{\dot{X}_{ML}-\dot{X}_{TL}}{(200)(100)}$	6	155
YDOT	$\frac{\dot{Y}_{ML}-\dot{Y}_{TL}}{200}$	$\frac{\dot{Y}_{ML}-\dot{Y}_{TL+}}{(200)(100)}$	7	156
ZDOT	$\frac{\dot{Z}_{ML}-\dot{Z}_{TL}}{200}$	$\frac{\dot{Z}_{ML}-\dot{Z}_{TL}}{(200)(100)}$	8	157

ANALOG VARIABLE NAME	AD/4 ANLG. VOLTAGE $-100<V<+100$	DIGITAL FRACTION $-1<N<+1$	CDC/6600 ADC ASSIGNMENT	AD/4 TRUNK LINE ASSIGNMENT
XXX	$\frac{K(X_{ML}-X_{TL})}{200}$	$\frac{K(X_{ML}-X_{TL})}{(200)(100)}$	9	170
YYY	$\frac{K(Y_{ML}-Y_{TL})}{200}$	$\frac{K(Y_{ML}-Y_{TL})}{(200)(100)}$	10	171
ZZZ	$\frac{K(Z_{ML}-Z_{TL})}{200}$	$\frac{K(Z_{ML}-Z_{TL})}{(200)(100)}$	12	173

TABLE 3.7
CDC/6600-AD/4 DAC ASSIGNMENTS

DIGITAL VARIABLE NAME	DIGITAL FRACTION $-1<N<+1$	AD/4 ANLG. VOLTAGE $-10<V<+10$	CDC/6600 DAC ASSIGNMENT	AD/4 TRUNK LINE ASSIGNMENT
XDTGO,XCOMP	$\frac{\dot{X}_{TL}}{(200)(10)}$	$\frac{\dot{X}_{TL}}{200}$	1	250
YDTGO,YCOMP	$\frac{\dot{Y}_{TL}}{(200)(10)}$	$\frac{\dot{Y}_{TL}}{200}$	2	251
ZDTGO,ZCOMP	$\frac{\dot{Z}_{TL}}{(200)(10)}$	$\frac{\dot{Z}_{TL}}{200}$	3	252
RLB	$\frac{(\ell/B)}{(10)}$	ℓ/B	4	253
COSE	$\frac{10\cos(\varepsilon)}{10}$	$10 \cos (\varepsilon)$	5	254
SPO	$\frac{T_{RP}}{10}$	T_{RP}	6	255
RI	$\frac{R_i}{(2000)(10)}$	$\frac{R_i}{2000}$	7	256

DIGITAL VARIABLE NAME	DIGITAL FRACTION -1 N +1	AD/4 ANLG. VOLTAGE -10 V +10	CDC/6600 DAC ASSIGNMENT	AD/4 TRUNK LINE ASSIGNMENT
GAM	$\frac{t'}{10}$	t'	8	257
EDOT	$\frac{(\gamma/t')}{(.6)(10)}$	$\frac{5(\gamma/t')}{3}$	9	270
THETAL	$\frac{\theta_L}{(10)(10)}$	$\frac{\theta_L}{10}$	10	271
RN	$\frac{10.741\ i_7}{10}$	$10.741\ i_7$	12	273
RN*	$\frac{R_n}{10}$	R_n	12	273

* FOR MICOM HYBRID APPLICATIONS

TABLE 3.8
AD/4-DIRECT CELL ADCs

ANALOG VARIABLE NAME	AD/4 ANALOG VOLTAGE	CDC/6600 ADC ASSIGNMENT	AD/4 TRUNK LINE ASSGN.
t_2	$9.53 \times 10^3 t^2$		220
t_4	$9.534 \times 10^3 t_4$		221
P_2	$9.896 \times 10^2 P_2$		222
R	.5555 R		223
P_1	1.7255 P_1		260
i_2	10.742 i_2		261
i_4	10.742 i_4		262
i_7	10.741 i_7		263
ψ_2	1.1111 ψ_2		320
θ_2	3.333 θ_2		321
ψ_4	1.111 ψ_4		322
θ_4	3.333 θ_4		323

ANALOG VARIABLE NAME	AD/4 ANALOG VOLTAGE	CDC/6600 ADC ASSIGNMENT	AD/4 TRUNK LINE ASSGN.
ψ_7	$1.111\ \psi_7$		360
θ_7	$3.333\ \theta_7$		361
P	1.25 P		362
Y	1.111 Y		363

TABLE 3.9
AD/4-SIBU INTERFACE
(COMMANDS)

ANALOG VARIABLE NAME	AD/4 ANALOG VOLTAGE	AD/4 TRUNK LINE ASSGN.	MASTER-PATCH CABLE CONN PIN/NO
r'	1.0 r'	350	10a-A/B
q'	1.0 q'	352	10a-C/D
p'	13.8×10^{-3}p'	354	10a-E/F
T_{RP}	.5555 T_{RP}	356	10a-G/H
$\dot{\theta}_7$	-.875 $\dot{\theta}_7$	370	10a-J/K
$\dot{\psi}_7$	-.875 $\dot{\psi}_7$	372	10a-L/M
$\dot{\theta}_2$	-.875 $\dot{\theta}_2$	374	10a-N/P
$\dot{\psi}_4$	+.875 $\dot{\psi}_4$	210	10b-A/B
$\dot{\theta}_4$	-.875 $\dot{\theta}_4$	212	10b-C/D
$\dot{\psi}_2$	-.875 $\dot{\psi}_2$	214	10b-E/F

TABLE 3.10
SIBU-AD/4 INTERFACE
(DATA)

ANALOG VARIABLE NAME	AD/4 ANALOG VOLTAGE	AD/4 TRUNK LINE ASSGN.	MASTER-PATCH CABLE CONN PIN/NO
r'	1.0 r'	216	A1B-G/H
q'	1.0 q'	230	A1B-J/K
p'	$13.8\text{x}10^{-3}$p'	232	A1B-L/M
$\delta_{\omega i}$	.5435 $\delta_{\omega i}$	010	A1B-A/B
Guid. Comd.	________	013	A1B-C/D
TAG	________	015	A1B-E/F
TAB	________	017	A1B-G/H
Sync Filt	________	031	A1B-J/K
Acquisition	________	033	A1B-L/M
Y	$56.9\text{x}10^{-3}$Y	20*	11A-A
P	$113.8\text{x}10^{-3}$P	24*	11A-E
R	$56.9\text{x}10^{-3}$R	22*	11A-C
Spare	$56.9\text{x}10^{-3}$R	23*	11A-D

* TRUNK ON EAI 429

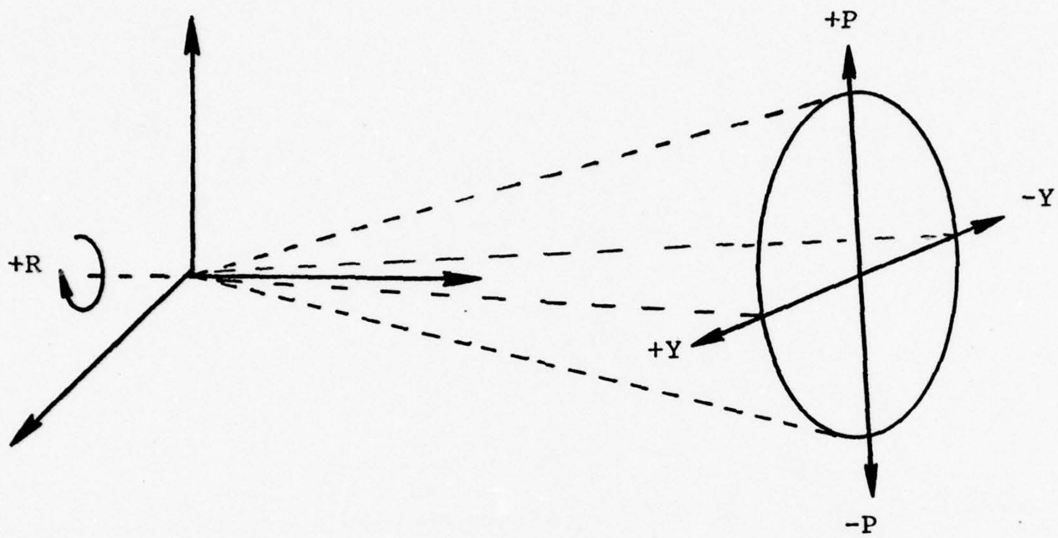
APPENDIX A

IRSS CALIBRATION LIMITS AND SIGN CONVENTIONS

APPENDIX A

IRSS CALIBRATION LIMITS AND SIGN CONVENTIONS

VARIABLE	CALIBRATION LIMITS	SIGN CONVENTIONS
θ_2	±30°	+θ is same as +P of the GUM
θ_4	±30°	+θ is same as +P of the GUM
θ_7	±30°	+θ is same as +P of the GUM
ψ_2	±90°	$+\psi_2$ is same as +Y of the GUM
ψ_4	±90°	$+\psi_4$ is same as +Y of the GUM
ψ_7	±90°	$+\psi_7$ is same as +Y of the GUM
Y	±90°	+Y is left (looking from rear to front of missile)
P	±80°	+P is up
R	±180°	+R is clockwise when viewing target from GUM
T_2	.02987-.00034 RAD	
P_1	57.955°-10.947°	increasing P_1 = increasing angle
P_2	.10105-0.0 RAD	increasing P_2 = increasing length
i_2	9.309-.004 VOLTS	increasing i_2 = increasing width
i_7	9.310-.004 VOLTS	increasing i_7 = increasing width
$\dot{P}$		
$\dot{Y}$		
$\dot{R}$		
TRP	±180°	+TRP is a counterclockwise rotation of the target as seen by the seeker



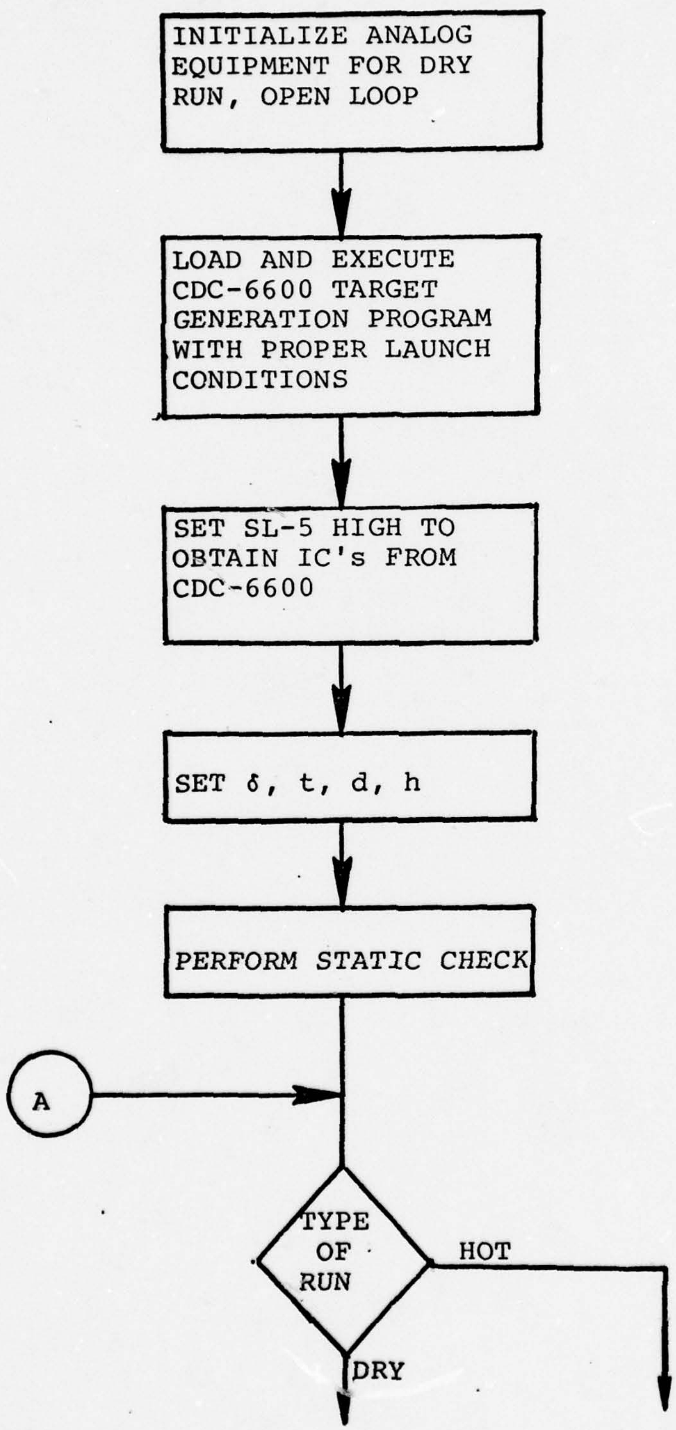
IRSS SIGN CONVENTIONS FOR
SEEKER ROLL, PITCH AND YAW

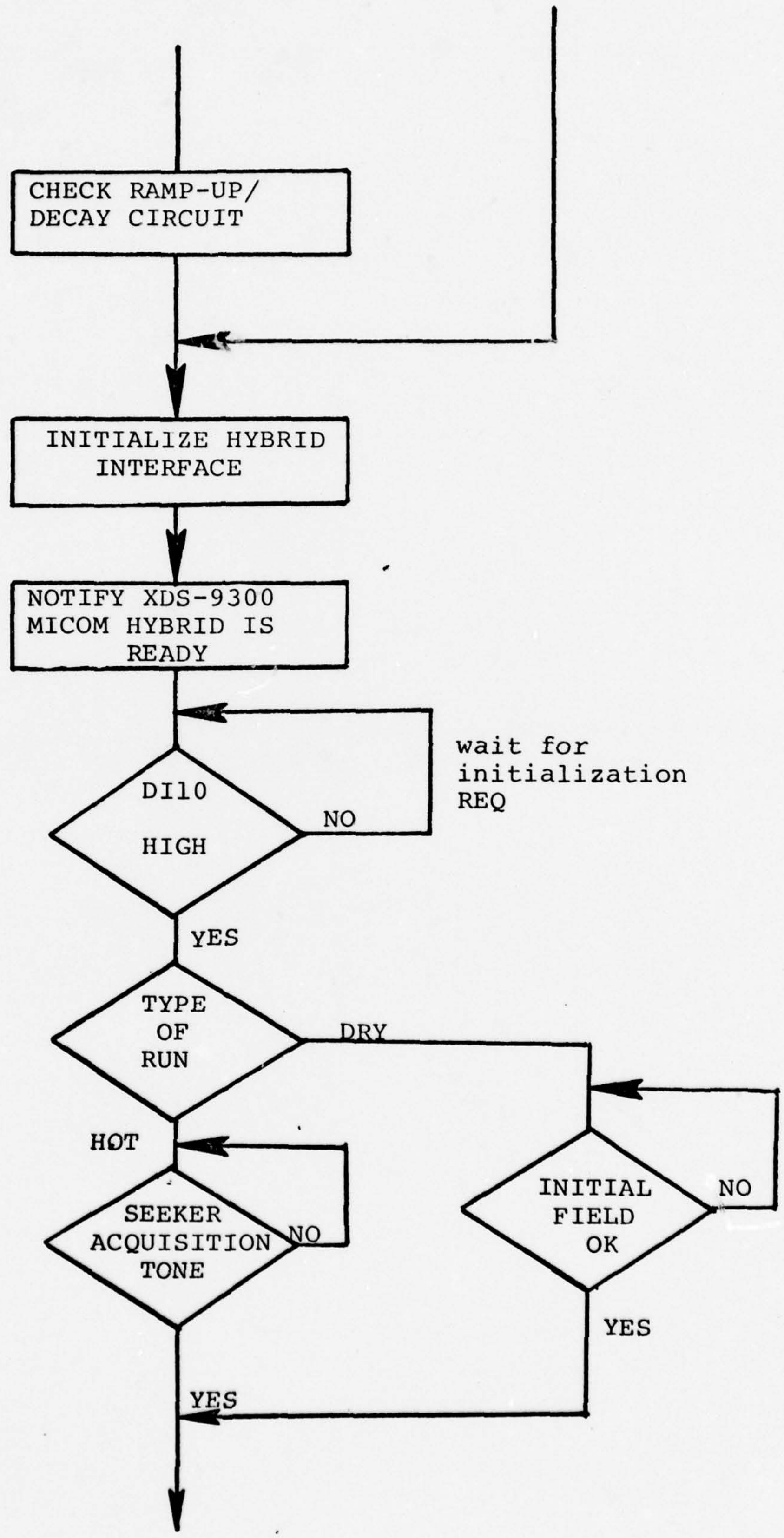
APPENDIX B

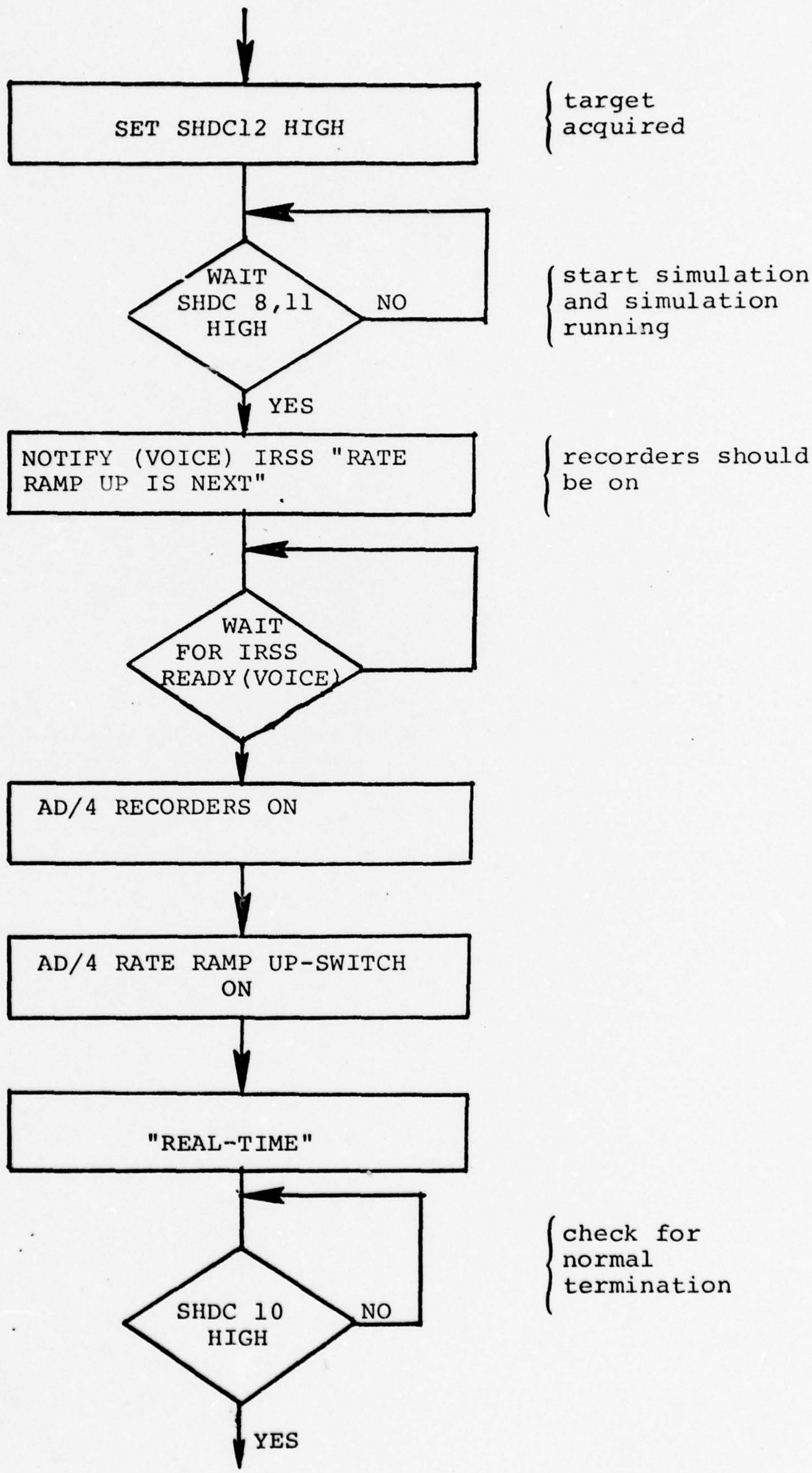
AD/4 FUNCTIONAL OPERATION SEQUENCE

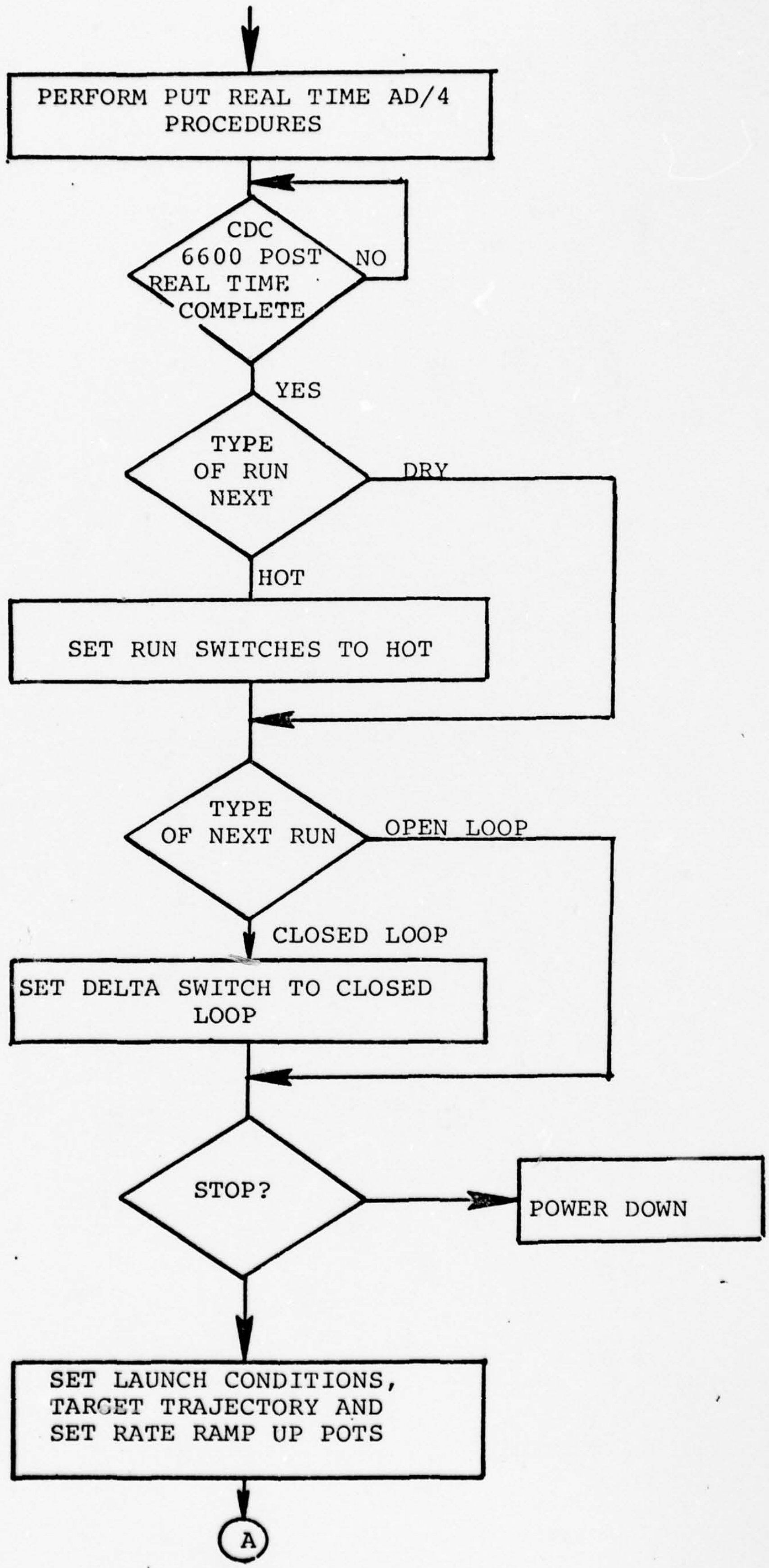
APPENDIX B

AD/4 FUNCTIONAL OPERATION SEQUENCE




CHECK RAMP-UP/ DECAY CIRCUIT
INITIALIZE HYBRID INTERFACE
NOTIFY XDS-9300 MICOM HYBRID IS READY
DI10 HIGH
NO
wait for initialization REQ
YES
TYPE OF RUN
DRY
HOT
SEEKER ACQUISITION TONE
NO
INITIAL FIELD OK
NO
YES
YES


SET SHDC12 HIGH
target acquired
WAIT SHDC 8,11 HIGH
NO
start simulation and simulation running
YES
NOTIFY (VOICE) IRSS "RATE RAMP UP IS NEXT"
recorders should be on
WAIT FOR IRSS READY (VOICE)
AD/4 RECORDERS ON
AD/4 RATE RAMP UP-SWITCH ON
"REAL-TIME"
SHDC 10 HIGH
NO
check for normal termination
YES


PERFORM PUT REAL TIME AD/4 PROCEDURES
CDC 6600 POST REAL TIME COMPLETE
NO
YES
TYPE OF RUN NEXT
DRY
HOT
SET RUN SWITCHES TO HOT
TYPE OF NEXT RUN
OPEN LOOP
CLOSED LOOP
SET DELTA SWITCH TO CLOSED LOOP
STOP?
POWER DOWN
SET LAUNCH CONDITIONS, TARGET TRAJECTORY AND SET RATE RAMP UP POTS
A

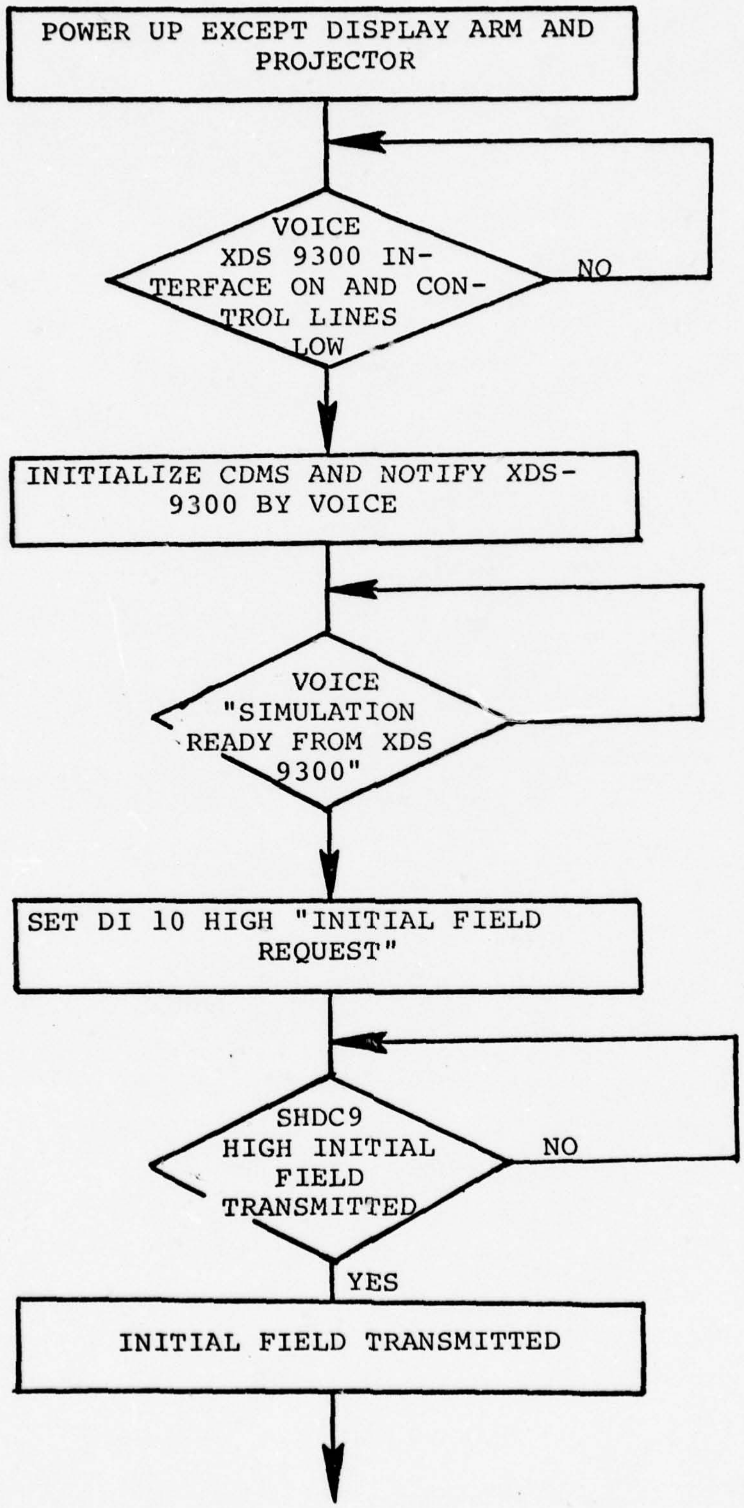
APPENDIX C

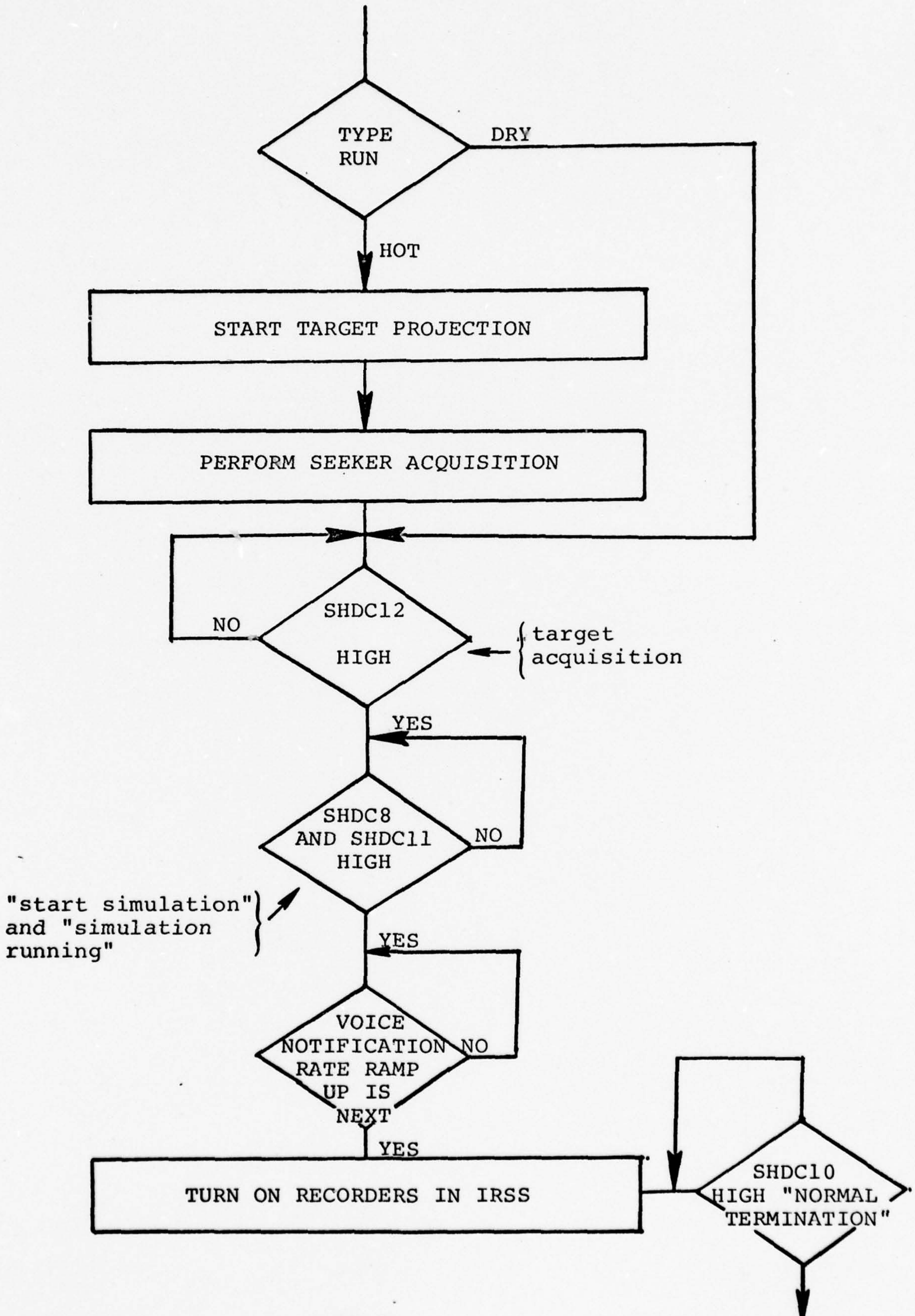
IRSS FUNCTIONAL OPERATION SEQUENCE

B&K DYNAMICS, INC.

APPENDIX C

IRSS FUNCTIONAL OPERATION SEQUENCE




TYPE RUN
DRY
HOT
START TARGET PROJECTION
PERFORM SEEKER ACQUISITION
SHDC12 HIGH
NO
target acquisition
YES
SHDC8 AND SHDC11 HIGH
NO
"start simulation" and "simulation running"
YES
VOICE NOTIFICATION RATE RAMP UP IS NEXT
NO
YES
TURN ON RECORDERS IN IRSS
SHDC10 HIGH "NORMAL TERMINATION"

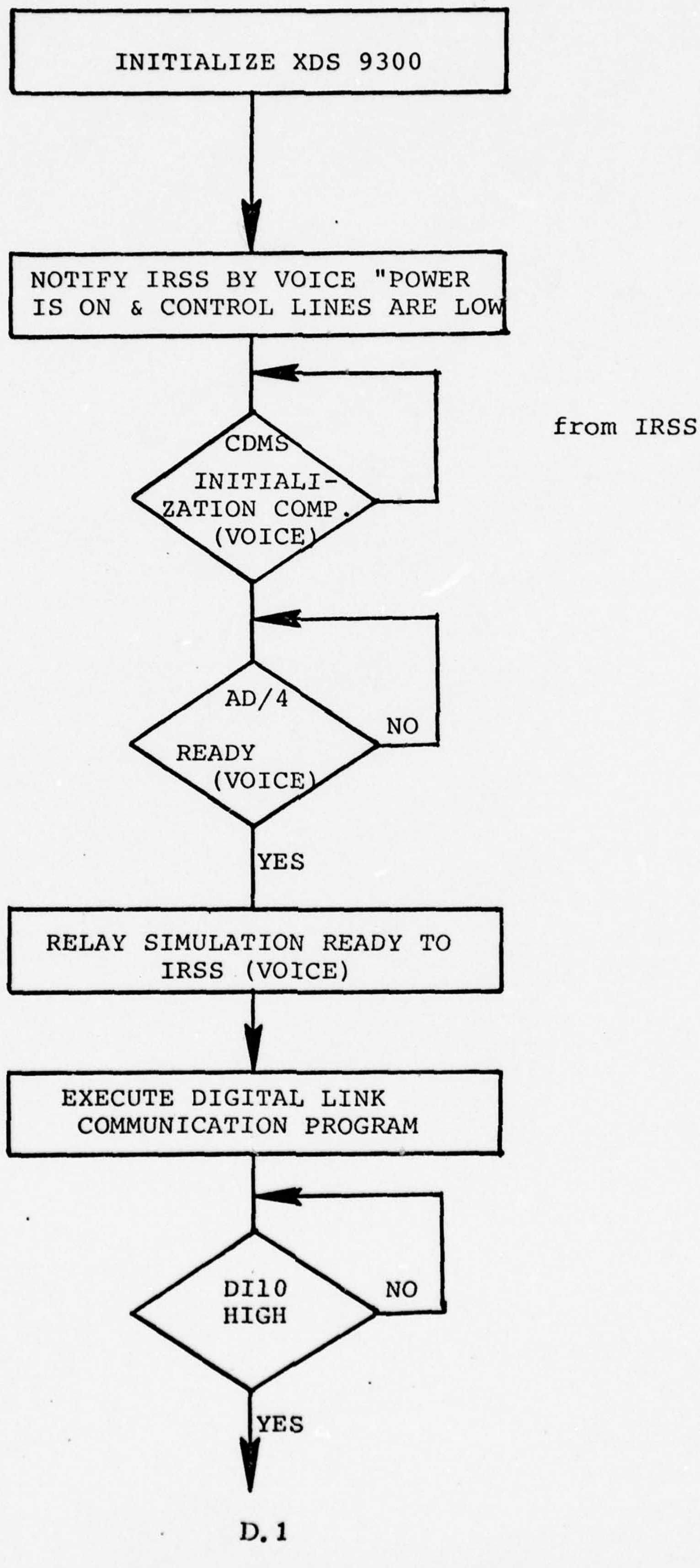
APPENDIX D

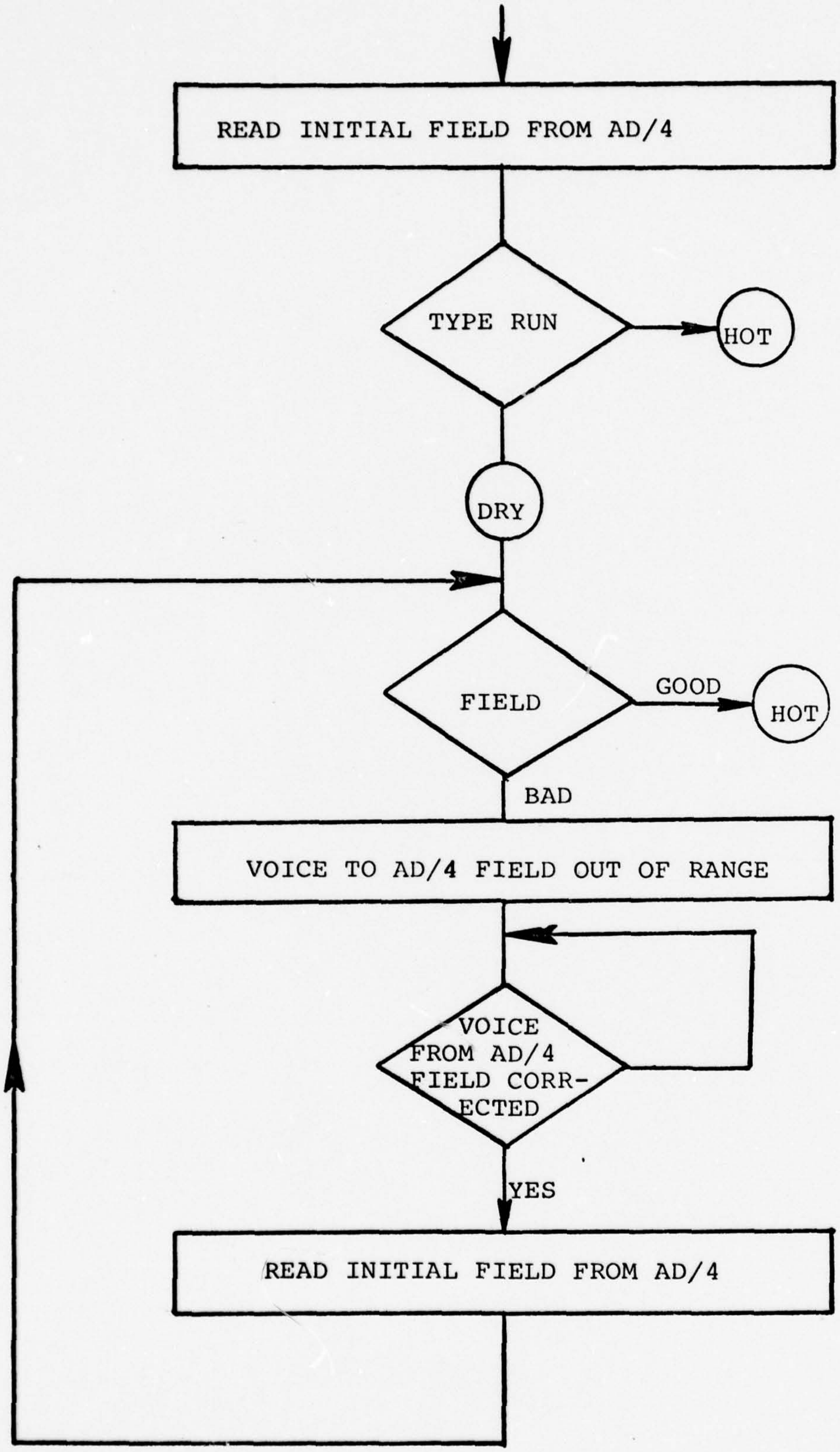
SDS/9300 FUNCTIONAL OPERATION SEQUENCE

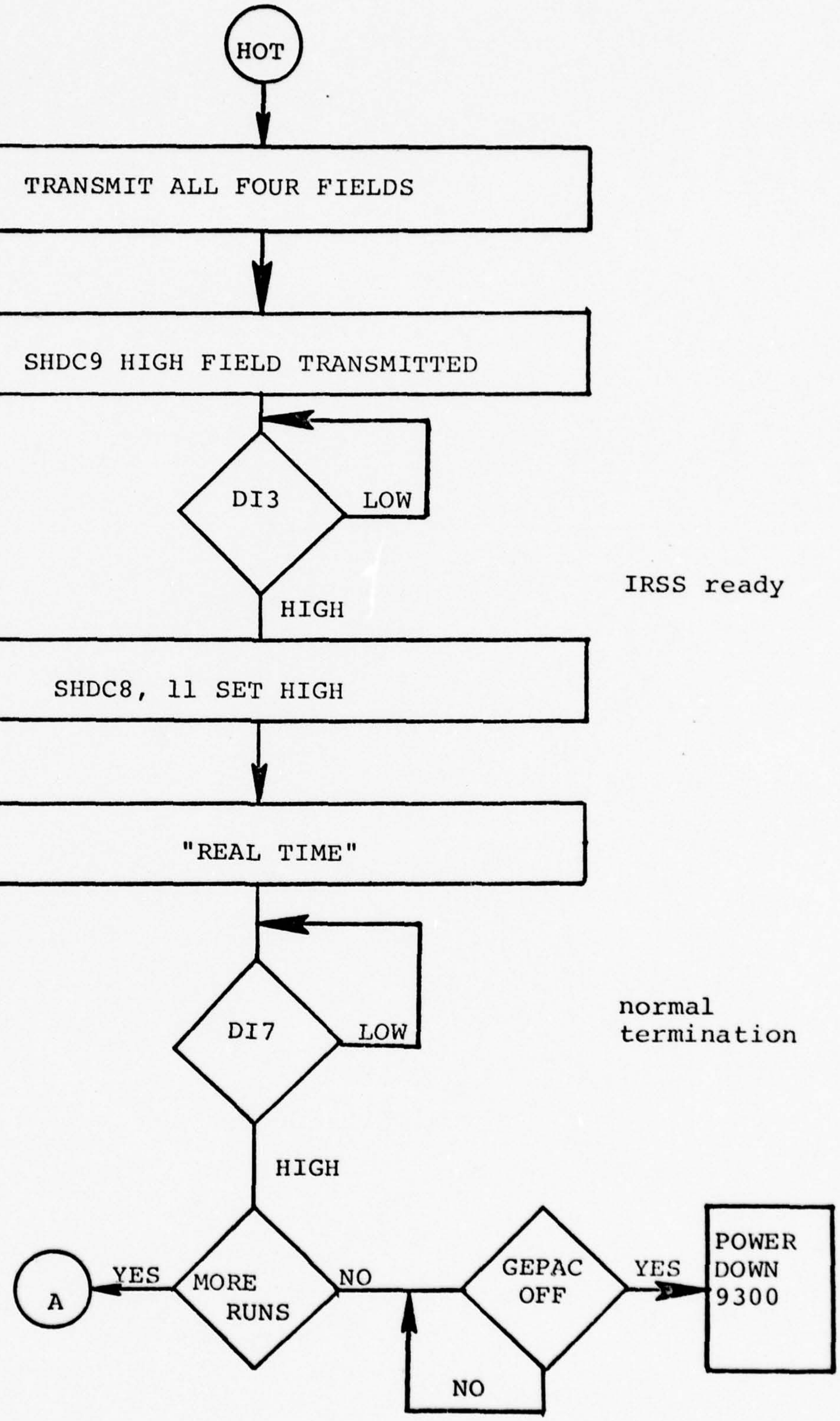
B&K DYNAMICS, INC.

APPENDIX D

SDS/9300 FUNCTIONAL OPERATION SEQUENCE




READ INITIAL FIELD FROM AD/4
TYPE RUN
HOT
DRY
FIELD
GOOD
HOT
BAD
VOICE TO AD/4 FIELD OUT OF RANGE
VOICE FROM AD/4 FIELD CORR-ECTED
YES
READ INITIAL FIELD FROM AD/4


HOT
TRANSMIT ALL FOUR FIELDS
SHDC9 HIGH FIELD TRANSMITTED
DI3
LOW
HIGH
IRSS ready
SHDC8, 11 SET HIGH
"REAL TIME"
DI7
LOW
normal
termination
HIGH
A
YES
MORE
RUNS
NO
GEPAC
OFF
YES
POWER
DOWN
9300
NO

APPENDIX E

DIRECT CELL

APPENDIX E

DIRECT CELL

INTRODUCTION

The direct cell concept incorporated into the ASC is an extension of the CDC 6600 which permits access to other equipment, inparticular, other computers.

There are two ways to do this:

1. Have the other computer do an input/output operation.
2. Reach directly into the memory, bypassing the CPU.

Most modern computers have the latter capability; it is generally called direct memory access or DMA.

To read or write data from or to a computer memory, the memory must be supplied with the following:

1. The address of the word to be referenced.
2. Control signal indicating which operation, read or write, is to be performed.
3. If a write, the data to be written.
4. Various timing and control signals which differ in detail from machine to machine.

THE SIGNAL CONDITIONER (S/C)

The signal conditioner is a black box designed to interface with the DMA of a specific computer on one side and with the CDC direct cell on the other. This hardware takes care of item 4 above: it provides signal level conversion, timing, and control signals to the DMA interface of the remote computer.

Thus, at MICOM, direct cell A can be used with an Interdata computer of the type used in the IRSS; B interfaces with a PDP11 as in the EOSS; and C interfaces with a SEL 8600 in the EAR.

THE ADDRESS MEMORY

Part of the direct cell consists of the address memory. This memory is loaded, under 6600 program control, with the actual addresses of the words to be read or written in the remote computer. Also in this memory are two control bits for each address; one specifies the direction, read or write, of the access; the other is used to conditionally control the access.

If the read/write bit is set, the remote memory is read; if the transmit address bit is set, the address will be sent to the remote computer.

THE DATA MEMORY

For each word of address memory there is a corresponding word of data memory; if the read/write bit of the address word is set the data memory word will receive the data read from the remote computer; if the bit is reset the current contents of the data word will be sent to the remote computer.

The address and data memories are 256 (expandable to 512) words long.

THE LIST MEMORY

Since it may not be desireable to read/write 256 (512) words from/to the remote computer, the direct cell provides a memory in which to store the length of the lists in Address & Data memories.

Also, it may be desireable to vary the address list at various points in the program. If all the desired lists can be held in the address/data memories we then need a pointer to the first word of each list. This pointer is stored in the List Memory along with the list length. The list memory thus provides a starting location and length in the address/data memories for the signal conditioner to process. The List Memory is 256 (expandable to 512) words long.

THE SEQUENCE MEMORY

In order to initiate a specific set of operations we must somehow inform the direct cell of the location of a list control word (in the list memory) which is to be processed. This would require 9 bits or 512 control lines. In order to reduce this number a sequence memory is provided.

The sequence memory is 8 (expandable to 16) words long. The contents of a particular sequence word is the address of a list memory word. An operation is initiated by specifying a sequence memory address (requires 4 bits or 16 lines); the list memory address is fetched from the designated sequence memory word; using this address, the list memory is accessed to retrieve the data and address memory starting location and length; these addresses and data are sent to the signal conditioner.

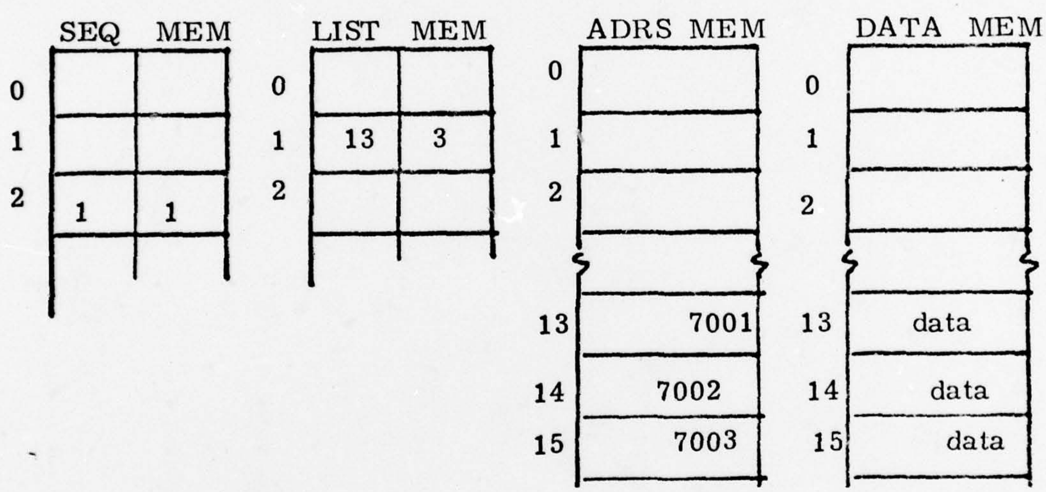
To provide further flexibility, the sequence memory also stores a list length; thus, the address/data memories form lists of remote memory references; the list memory delineates these lists; and the sequence memory defines lists of lists.

The entire process is initiated by an interrupt: interrupt *n* fetches the contents of sequence memory word *n* and the process then continues until all the referenced lists are exhausted.

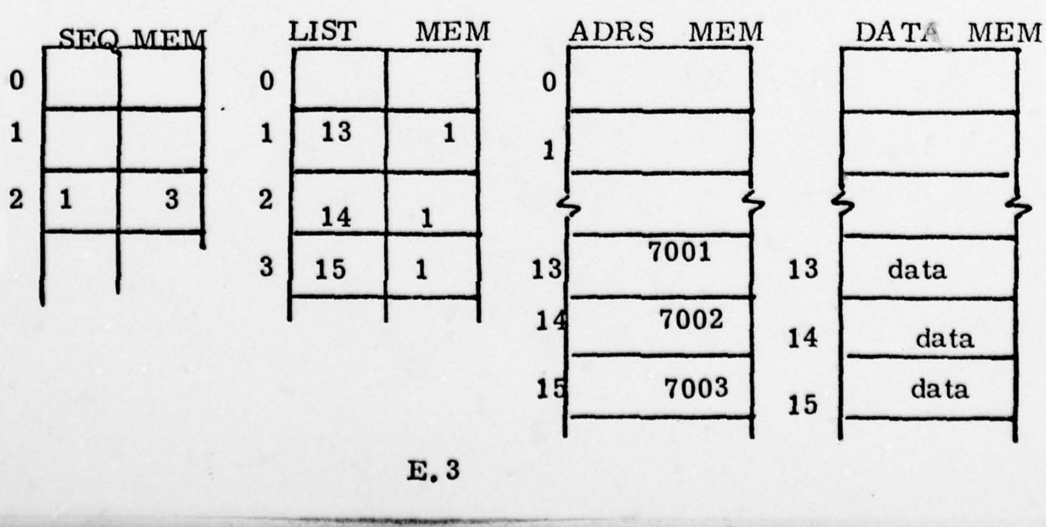
Examples

1) Interrupt 2 is to initiate a 3 word transfer at addresses 7001, 7002 and 7003 in the remote computer.

Method A



Method B



In method A word 2 of sequence memory (associated with interrupt 2) specifies an address of 1 and a length of 1 in the list memory: thus there is one master list. Word 1 of list memory specifies a 3 word list starting at word 13 of address/data memory: thus, there is one list, 3 words long. Words 13, 14, and 15 of address memory specify the remote memory locations to be accessed; the corresponding words of data memory provide/receive the actual data.

2) Interrupt 0 is to initiate a 2 word transfer at addresses 5001 and 5002; interrupt 1 is to initiate a 3 word transfer at addresses 6001, 6002 and 6003; interrupt 2 is to perform both transfers.

SEQ MEM

0	0	1
1	1	1
2	0	2

LIST MEM

0	0	2
1	2	3

ADRS MEM

0	5001
1	5002
2	6001
3	6002
4	6003

DATA MEM

0	data
1	data
2	data
3	data
4	data

DATA MEMORY FORMAT

For a write operation at the remote computer, the data memory must be loaded by the 6600. Each data memory cell has two addresses by which the 6600 can load it. One address simply loads the 6600 data into the requisite data memory word; the other address routes the data through an unpacking circuit, thus converting to integer format from floating point.

For a read operation from the remote computer, the 6600 must be able to unload the data memory. Each data memory cell has two addresses by which the 6600 may read it; one address simply passes the data through to the 6600; the other routes it through a packing circuit which converts to floating point from integer.

Note that the 6600 uses one's couplement for negative numbers, while most mini-computers use two's complement.

The data memory is 16 bits wide, expandable to 32.

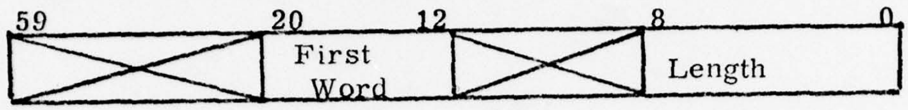
ADDRESS MEMORY FORMAT

The address memory contains the actual remote memory address in bits 0-16; the read/write bit is in bit 24; and the transmit address bit is in bit 25.

59 — 26	25	24	23 — 17	16 — 0
	TA	R/W		Remote ADRS

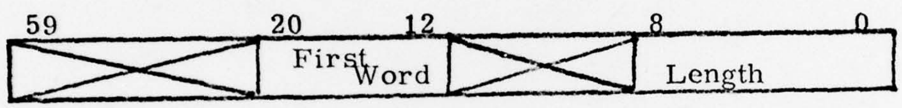
LIST MEMORY FORMAT

The data length is in bits 0-8; the data address is in bits 12-20.



SEQUENCE MEMORY FORMAT

The list length is in bits 0-8; the list address is in bits 12-20.



ADDRESSING THE DIRECT CELL

The direct cell is interfaced to the 6600 central memory bus. In the 6600, addresses may be 18 bits in length (2^{18} = 256K words maximum memory). However, only 17 bits are used for actual memory addresses; any address larger than $2^{17} - 1$ is ignored by the central memory, but recognized by the memory bus adapter. Hence an attempt to reference an address larger than $2^{17} - 1$ does two things: a) disables the automatic addition of a base address and b) actually references equipment attached to the memory bus adapter. The actual hardware referenced is a function of the interconnection to the adapter.

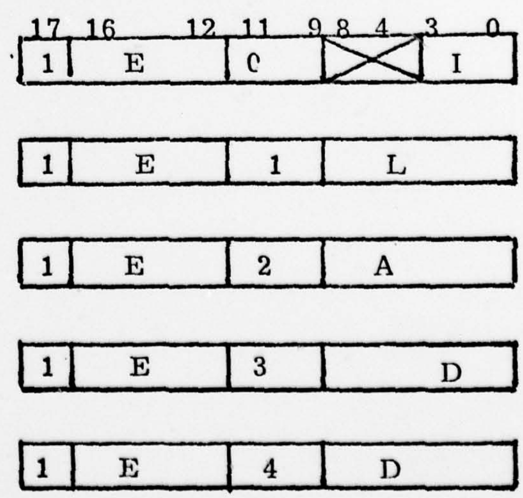
SEQUENCE MEMORY ADDRESS

LIST MEMORY ADDRESS

ADDRESS MEMORY ADDRESS

DATA MEMORY ADDRESS (F. P.)

DATA MEMORY ADDRESS (Int)



E = Equipment code (thumbwheels on cabinets)
I = Interrupt number (also sequence memory address)
L = List memory address
A = Address memory address
D = Data memory address

ACQUISITION REGISTERS

Associated with each interrupt (and therefore each word of sequence memory) are two 20 bit acquisition registers: the static and the dynamic.

Upon the receipt of an interrupt, the static register is copied into the dynamic register. The dynamic register is decremented every 10 μs. If the transfer associated with the interrupt are successfully terminated before the dynamic register hits zero, the decrementing is halted. Thus, the dynamic register acts as a "time out" register: if it never hit zero the transfers occured within the allotted time. The static register holds the maximum allotted time.

STATIC ACQUISITION REGISTER ADDRESS

17		11	9	3 0
1	E	6	O	I

STATIC ACQUISITION REGISTER DATA

	19 0
O	XXX

Writing into the static acquisition register may not be done by a central processor in the program mode unless the program mode setup enable flag is set.

The acquisition registers may be read in program mode.

STATIC ACQUISITION REGISTER ADDRESS

17		11	9	3 0
1	E	5	O	I

DYNAMIC ACQUISITION REGISTER ADDRESS

1	E	5	1	I

INTERRUPT TIMING

If the direct cell is inactive upon receipt of an interrupt, the transfer begins immediately. If it is active, the interrupt number is placed in a first in, first out list. When the current transfer is complete, the interrupt at the head of the list is initiated.

The dynamic acquisition registers of all interrupts in the list are decremented, thus measuring total elapsed time from receipt of the interrupt to completion of the transfer.

PROGRAM INITIATIVE OF TRANSFER

An address is provided to initiate an interrupt by programming. The data word sent to the direct cell has a bit set for each interrupt to be pulsed. Bit n set causes interrupt n to be pulsed where 0 is less than or equal to n is less than or equal to 15.

DIRECT CELL STROBE ADDRESS

17	16 — 12	11 — 9		4 — 0
1	E	5	O	20

STROBE DATA

59 — 16	15 — 0
	XXXX . . . X

MONITORING CELL ACTIVITY

The activity register can be read at any time. This provides the address of the current or last address/data memories; the current/last list memory address; and the current/list interrupt (sequence memory address).

ACTIVITY REGISTER ADDRESS

17	16 — 12	8	5	4 — 0
1	E	5	O	21

DATA

	27 — 24		20 — 12		8 — 0
	INTRRT		LIST ADD		WORD ADD

APPENDIX F

BASIC INSTRUCTION SET

B&K DYNAMICS, INC.

BASIC INSTRUCTION SET

The following section presents the basic instruction set of the 8008.

A. Data and Instruction Formats

Data in the 8008 is stored in the form of 8-bit binary integers. All data transfers to the system data bus will be in the same format.

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

DATA WORD

The program instructions may be one, two, or three bytes in length. Multiple byte instructions must be stored in successive words in program memory. The instruction formats then depend on the particular operation executed.

Format		TYPICAL INSTRUCTIONS
One Byte Instructions		
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OP CODE	Register to register, memory reference, I/O arithmetic or logical, rotate or return instructions
Two Byte Instructions		
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OP CODE	
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OPERAND	Immediate mode instructions
Three Byte Instructions		
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OP CODE	
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	LOW ADDRESS	JUMP or CALL instructions
X X D_5 D_4 D_3 D_2 D_1 D_0	HIGH ADDRESS*	

*For the third byte of this instruction, D_6 and D_7 are "don't care" bits.

For the MCS-8 a logic "1" is defined as a high level and a logic "0" is defined as a low level.

B. Summary of Processor Instructions

Index Register Instructions

The load instructions do not affect the flag flip-flops. The increment and decrement instructions affect all flip-flops except the carry.

MNEMONIC	MINIMUM STATES REQUIRED	INSTRUCTION CODE D_7 D_6	D_5 D_4 D_3	D_2 D_1 D_0	DESCRIPTION OF OPERATION
(1) Lr_1r_2	(5)	1 1	D D D	S S S	Load index register r_1 with the content of index register r_2.
(2) LrM	(8)	1 1	D D D	1 1 1	Load index register r with the content of memory register M.
LMr	(7)	1 1	1 1 1	S S S	Load memory register M with the content of index register r.
(3) LrI	(8)	0 0	D D D	1 1 0	Load index register r with data B . . . B.
		B B	B B B	B B B	
LMI	(9)	0 0	1 1 1	1 1 0	Load memory register M with data B . . . B.
		B B	B B B	B B B	
INr	(5)	0 0	D D D	0 0 0	Increment the content of index register r ($r \neq A$).
DCr	(5)	0 0	D D D	0 0 1	Decrement the content of index register r ($r \neq A$).

Accumulator Group Instructions

The result of the ALU instructions affect all of the flag flip-flops. The rotate instructions affect only the carry flip-flop.

ADr	(5)	1 0	0 0 0	S S S	Add the content of index register r, memory register M, or data B . . . B to the accumulator. An overflow (carry) sets the carry flip-flop.
ADM	(8)	1 0	0 0 0	1 1 1	
ADI	(8)	0 0	0 0 0	1 0 0	
		B B	B B B	B B B	
ACr	(5)	1 0	0 0 1	S S S	Add the content of index register r, memory register M, or data B . . . B to the accumulator with carry. An overflow (carry) sets the carry flip-flop.
ACM	(8)	1 0	0 0 1	1 1 1	
ACI	(8)	0 0	0 0 1	1 0 0	
		B B	B B B	B B B	
SUr	(5)	1 0	0 1 0	S S S	Subtract the content of index register r, memory register M, or data B . . . B from the accumulator. An underflow (borrow) sets the carry flip-flop.
SUM	(8)	1 0	0 1 0	1 1 1	
SUI	(8)	0 0	0 1 0	1 0 0	
		B B	B B B	B B B	
SBr	(5)	1 0	0 1 1	S S S	Subtract the content of index register r, memory register M, or data B . . . B from the accumulator with borrow. An underflow (borrow) sets the carry flip-flop.
SBM	(8)	1 0	0 1 1	1 1 1	
SBI	(8)	0 0	0 1 1	1 0 0	
		B B	B B B	B B B	

MNEMONIC	MINIMUM STATES REQUIRED	INSTRUCTION CODE $D_7 D_6$	$D_5 D_4 D_3$	$D_2 D_1 D_0$	DESCRIPTION OF OPERATION
NDr	(5)	1 0	1 0 0	S S S	Compute the logical AND of the content of index register r, memory register M, or data B . . . B with the accumulator.
NDM	(8)	1 0	1 0 0	1 1 1	
NDI	(8)	0 0 B B	1 0 0 B B B	1 0 0 B B B	
XRr	(5)	1 0	1 0 1	S S S	Compute the EXCLUSIVE OR of the content of index register r, memory register M, or data B . . . B with the accumulator.
XRM	(8)	1 0	1 0 1	1 1 1	
XRI	(8)	0 0 B B	1 0 1 B B B	1 0 0 B B B	
ORr	(5)	1 0	1 1 0	S S S	Compute the INCLUSIVE OR of the content of index register r, memory register m, or data B . . . B with the accumulator .
ORM	(8)	1 0	1 1 0	1 1 1	
ORI	(8)	0 0 B B	1 1 0 B B B	1 0 0 B B B	
CPr	(5)	1 0	1 1 1	S S S	Compare the content of index register r, memory register M, or data B . . . B with the accumulator. The content of the accumulator is unchanged.
CPM	(8)	1 0	1 1 1	1 1 1	
CPI	(8)	0 0 B B	1 1 1 B B B	1 0 0 B B B	
RLC	(5)	0 0	0 0 0	0 1 0	Rotate the content of the accumulator left.
RRC	(5)	0 0	0 0 1	0 1 0	Rotate the content of the accumulator right.
RAL	(5)	0 0	0 1 0	0 1 0	Rotate the content of the accumulator left through the carry.
RAR	(5)	0 0	0 1 1	0 1 0	Rotate the content of the accumulator right through the carry.

Program Counter and Stack Control Instructions

(4) JMP	(11)	0 1 B_2 B_2 X X	X X X B_2 B_2 B_2 B_3 B_3 B_3	1 0 0 B_2 B_2 B_2 B_3 B_3 B_3	Unconditionally jump to memory address $B_3 \ldots B_3 B_2 \ldots B_2$.
(5) JFc	(9 or 11)	0 1 B_2 B_2 X X	0 C_4 C_3 B_2 B_2 B_2 B_3 B_3 B_3	0 0 0 B_2 B_2 B_2 B_3 B_3 B_3	Jump to memory address $B_3 \ldots B_3 B_2 \ldots B_2$ if the condition flip-flop c is false. Otherwise, execute the next instruction in sequence.
JTc	(9 or 11)	0 1 B_2 B_2 X X	1 C_4 C_3 B_2 B_2 B_2 B_3 B_3 B_3	0 0 0 B_2 B_2 B_2 B_3 B_3 B_3	Jump to memory address $B_3 \ldots B_3 B_2 \ldots B_2$ if the condition flip-flop c is true. Otherwise, execute the next instruction in sequence.
CAL	(11)	0 1 B_2 B_2 X X	X X X B_2 B_2 B_2 B_3 B_3 B_3	1 1 0 B_2 B_2 B_2 B_3 B_3 B_3	Unconditionally call the subroutine at memory address $B_3 \ldots B_3 B_2 \ldots B_2$. Save the current address (up one level in the stack).
CFc	(9 or 11)	0 1 B_2 B_2 X X	0 C_4 C_3 B_2 B_2 B_2 B_3 B_3 B_3	0 1 0 B_2 B_2 B_2 B_3 B_3 B_3	Call the subroutine at memory address $B_3 \ldots B_3 B_2 \ldots B_2$ if the condition flip-flop c is false, and save the current address (up one level in the stack.) Otherwise, execute the next instruction in sequence.
CTc	(9 or 11)	0 1 B_2 B_2 X X	1 C_4 C_3 B_2 B_2 B_2 B_3 B_3 B_3	0 1 0 B_2 B_2 B_2 B_3 B_3 B_3	Call the subroutine at memory address $B_3 \ldots B_3 B_2 \ldots B_2$ if the condition flip-flop c is true, and save the current address (up one level in the stack). Otherwise, execute the next instruction in sequence.
RET	(5)	0 0	X X X	1 1 1	Unconditionally return (down one level in the stack).
RFc	(3 or 5)	0 0	0 C_4 C_3	0 1 1	Return (down one level in the stack) if the condition flip-flop c is false. Otherwise, execute the next instruction in sequence.
RTc	(3 or 5)	0 0	1 C_4 C_3	0 1 1	Return (down one level in the stack) if the condition flip-flop c is true. Otherwise, execute the next instruction in sequence.
RST	(5)	0 0	A A A	1 0 1	Call the subroutine at memory address AAA000 (up one level in the stack).

Input/Output Instructions

INP	(8)	0 1	0 0 M	M M 1	Read the content of the selected input port (MMM) into the accumulator.
OUT	(6)	0 1	R R M	M M 1	Write the content of the accumulator into the selected output port (RRMMM, RR ≠ 00).

Machine Instruction

HLT	(4)	0 0	0 0 0	0 0 X	Enter the STOPPED state and remain there until interrupted.
HLT	(4)	1 1	1 1 1	1 1 1	Enter the STOPPED state and remain there until interrupted.

NOTES:

(1) SSS = Source Index Register, DDD = Destination Index Register — These registers, r_i, are designated A(accumulator–000), B(001), C(010), D(011), E(100), H(101), L(110).

(2) Memory registers are addressed by the contents of registers H & L.

(3) Additional bytes of instruction are designated by BBBBBBBB.

(4) X = "Don't Care".

(5) Flag flip-flops are defined by $C_4 C_3$: carry (00-overflow or underflow), zero (01-result is zero), sign (10-MSB of result is "1"), parity (11-parity is even).

C. Complete Functional Definition

The following pages present a detailed description of the complete 8008 Instruction Set.

Symbols	Meaning
<B2>	Second byte of the instruction
<B3>	Third byte of the instruction
r	One of the scratch pad register references: A, B, C, D, E, H, L
c	One of the following flag flip-flop references: C, Z, S, P
C_4C_3	Flag flip-flop codes / Condition for True 00 carry — Overflow, underflow 01 zero — Result is zero 10 sign — MSB of result is "1" 11 parity — Parity of result is even
M	Memory location indicated by the contents of registers H and L
( )	Contents of location or register
$\wedge$	Logical product
$\veebar$	Exclusive "or"
$\vee$	Inclusive "or"
A_m	Bit m of the A-register
STACK	Instruction counter (P) pushdown register
P	Program Counter
$\leftarrow$	Is transferred to
XXX	A "don't care"
SSS	Source register for data
DDD	Destination register for data Register # (SSS or DDD) — Register Name 000 — A 001 — B 010 — C 011 — D 100 — E 101 — H 110 — L

INDEX REGISTER INSTRUCTIONS

LOAD DATA TO INDEX REGISTFRS – One Byte

Data may be loaded into or moved between any of the index registers, or memory registers.

Lr_1r_2 (one cycle – PCI)	11	DDD	SSS	$(r_1) \leftarrow (r_2)$ Load register r_1 with the content of r_2. The content of r_2 remains unchanged. If SSS=DDD, the instruction is a NOP (no operation).
LrM (two cycles – PCI/PCR)	11	DDD	111	$(r) \leftarrow (M)$ Load register r with the content of the memory location addressed by the contents of registers H and L. (DDD≠111 – HALT instr.)
LMr (two cycles – PCI/PCW)	11	111	SSS	$(M) \leftarrow (r)$ Load the memory location addressed by the contents of registers H and L with the content of register r. (SSS≠111 – HALT instr.)

LOAD DATA IMMEDIATE – Two Bytes

A byte of data immediately following the instruction may be loaded into the processor or into the memory

LrI (two cycles – PCI/PCR)	00	DDD <B_2>	110	$(r) \leftarrow \langle B_2 \rangle$ Load byte two of the instruction into register r.
LMI (three cycles – PCI/PCR/PCW)	00	111 <B_2>	110	$(M) \leftarrow \langle B_2 \rangle$ Load byte two of the instruction into the memory location addressed by the contents of registers H and L.

INCREMENT INDEX REGISTER – One Byte

INr (one cycle – PCI)	00	DDD	000	$(r) \leftarrow (r)+1$. The content of register r is incremented by one. All of the condition flip-flops except carry are affected by the result. Note that DDD≠000 (HALT instr.) and DDD≠111 (content of memory may not be incremented).

DECREMENT INDEX REGISTER – One Byte

DCr (one cycle – PCI)	00	DDD	001	$(r) \leftarrow (r)-1$. The content of register r is decremented by one. All of the condition flip-flops except carry are affected by the result. Note that DDD≠000 (HALT instr.) and DDD≠111 (content of memory may not be decremented).

ACCUMULATOR GROUP INSTRUCTIONS

Operations are performed and the status flip-flops, C, Z, S, P, are set based on the result of the operation. Logical operations (NDr, XRr, ORr) set the carry flip-flop to zero. Rotate operations affect only the carry flip-flop. Two's complement subtraction is used.

ALU INDEX REGISTER INSTRUCTIONS – One Byte

(one cycle – PCI)

Index Register operations are carried out between the accumulator and the content of one of the index registers (SSS=000 thru SSS=110). The previous content of register SSS is unchanged by the operation.

ADr	10	000	SSS	$(A) \leftarrow (A)+(r)$ Add the content of register r to the content of register A and place the result into register A.
ACr	10	001	SSS	$(A) \leftarrow (A)+(r)+(\text{carry})$ Add the content of register r and the contents of the carry flip-flop to the content of the A register and place the result into Register A.
SUr	10	010	SSS	$(A) \leftarrow (A)-(r)$ Subtract the content of register r from the content of register A and place the result into register A. Two's complement subtraction is used.

ACCUMULATOR GROUP INSTRUCTIONS - Cont'd.

SBr	10	011	SSS	(A)←(A)−(r)−(borrow) Subtract the content of register r and the content of the carry flip flop from the content of register A and place the result into register A.
NDr	10	100	SSS	(A)←(A)∧(r) Place the logical product of the register A and register r into register A.
XRr	10	101	SSS	(A)←(A)∀(r) Place the "exclusive - or" of the content of register A and register r into register A.
ORr	10	110	SSS	(A)←(A)V(r) Place the "inclusive - or" of the content of register A and register r into register A.
CPr	10	111	SSS	(A)−(r) Compare the content of register A with the content of register r. The content of register A remains unchanged. The flag flip-flops are set by the result of the subtraction. Equality (A=r) is indicated by the zero flip-flop set to "1". Less than (A<r) is indicated by the carry flip-flop, set to "1".

ALU OPERATIONS WITH MEMORY – One Byte
(two cycles – PCI/PCR)
Arithmetic and logical operations are carried out between the accumulator and the byte of data addressed by the contents of registers H and L.

ADM	10	000	111	(A)←(A)+(M) ADD
ACM	10	001	111	(A)←(A)+(M)+(carry) ADD with carry
SUM	10	010	111	(A)←(A)−(M) SUBTRACT
SBM	10	011	111	(A)←(A)−(M)−(borrow) SUBTRACT with borrow
NDM	10	100	111	(A)←(A)∧(M) Logical AND
XRM	10	101	111	(A)←(A)∀(M) Exclusive OR
ORM	10	110	111	(A)←(A)V(M) Inclusive OR
CPM	10	111	111	(A)−(M) COMPARE

ALU IMMEDIATE INSTRUCTIONS – Two Bytes
(two cycles –PCI/PCR)
Arithmetic and logical operations are carried out between the accumulator and the byte of data immediately following the instruction.

ADI	00	000 <B_2>	100	(A)←(A)+<B_2> ADD
ACI	00	001 <B_2>	100	(A)←(A)+<B_2>+(carry) ADD with carry
SUI	00	010 <B_2>	100	(A)←(A)−<B_2> SUBTRACT
SBI	00	011 <B_2>	100	(A)←(A)−<B_2>−(borrow) SUBTRACT with borrow
NDI	00	100 <B_2>	100	(A)←(A)∧<B_2> Logical AND
XRI	00	101 <B_2>	100	(A)←(A)∀<B_2> Exclusive OR
ORI	00	110 <B_2>	100	(A)←(A)V<B_2> Inclusive OR
CPI	00	111 <B_2>	100	(A)−<B_2> COMPARE

ROTATE INSTRUCTIONS – One Byte
(one cycle – PCI)
The accumulator content (register A) may be rotated either right or left, around the carry bit or through the carry bit. Only the carry flip-flop is affected by these instructions; the other flags are unchanged.

RLC	00 000 010	$A_{m+1} \leftarrow A_m$, $A_0 \leftarrow A_7$, (carry) $\leftarrow A_7$ Rotate the content of register A left one bit. Rotate A_7 into A_0 and into the carry flip-flop.
RRC	00 001 010	$A_m \leftarrow A_{m+1}$, $A_7 \leftarrow A_0$, (carry) $\leftarrow A_0$ Rotate the content of register A right one bit. Rotate A_0 into A_7 and into the carry flip-flop.
RAL	00 010 010	$A_{m+1} \leftarrow A_m$, $A_0 \leftarrow$ (carry), (carry) $\leftarrow A_7$ Rotate the content of Register A left one bit. Rotate the content of the carry flip-flop into A_0. Rotate A_7 into the carry flip-flop.
RAR	00 011 010	$A_m \leftarrow A_{m+1}$, $A_7 \leftarrow$ (carry), (carry) $\leftarrow A_0$ Rotate the content of register A right one bit. Rotate the content of the carry flip-flop into A_7. Rotate A_0 into the carry flip-flop.

PROGRAM COUNTER AND STACK CONTROL INSTRUCTIONS

JUMP INSTRUCTIONS – Three Bytes
(three cycles – PCI/PCR/PCR)
Normal flow of the microprogram may be altered by jumping to an address specified by bytes two and three of an instruction.

JMP (Jump Unconditionally)	01 XXX 100 $\langle B_2 \rangle$ $\langle B_3 \rangle$	(P) $\leftarrow \langle B_3 \rangle \langle B_2 \rangle$ Jump unconditionally to the instruction located in memory location addressed by byte two and byte three.
JFc (Jump if Condition False)	01 $0C_4C_3$ 000 $\langle B_2 \rangle$ $\langle B_3 \rangle$	*If (c) = 0, (P) $\leftarrow \langle B_3 \rangle \langle B_2 \rangle$. Otherwise, (P) = (P)+3.* If the content of flip-flop c is zero, then jump to the instruction located in memory location $\langle B_3 \rangle \langle B_2 \rangle$; otherwise, execute the next instruction in sequence.
JTc (Jump if Condition True)	01 $1C_4C_3$ 000 $\langle B_2 \rangle$ $\langle B_3 \rangle$	If (c) = 1, (P) $\leftarrow \langle B_3 \rangle \langle B_2 \rangle$. Otherwise, (P) = (P)+3. If the content of flip-flop c is one, then jump to the instruction located in memory location $\langle B_3 \rangle \langle B_2 \rangle$; otherwise, execute the next instruction in sequence.

CALL INSTRUCTIONS – Three Bytes
(three cycles – PCI/PCR/PCR)
Subroutines may be called and nested up to seven levels.

CAL (Call subroutine Unconditionally)	01 XXX 110 $\langle B_2 \rangle$ $\langle B_3 \rangle$	(Stack) $\leftarrow$ (P), (P) $\leftarrow \langle B_3 \rangle \langle B_2 \rangle$. Shift the content of P to the pushdown stack. Jump unconditionally to the instruction located in memory location addressed by byte two and byte three.
CFc (Call subroutine if Condition False)	01 $0C_4C_3$ 010 $\langle B_2 \rangle$ $\langle B_3 \rangle$	If (c) = 0, (Stack) $\leftarrow$ (P), (P) $\leftarrow \langle B_3 \rangle \langle B_2 \rangle$. Otherwise, (P) = (P)+3. If the content of flip-flop c is zero, then shift contents of P to the pushdown stack and jump to the instruction located in memory location $\langle B_3 \rangle \langle B_2 \rangle$; otherwise, execute the next instruction in sequence.
CTc (Call subroutine if Condition True)	01 $1C_4C_3$ 010 $\langle B_2 \rangle$ $\langle B_3 \rangle$	If (c) = 1, (Stack) $\leftarrow$ (P), (P) $\leftarrow \langle B_3 \rangle \langle B_2 \rangle$. Otherwise, (P) = (P)+3. If the content of flip-flop c is one, then shift contents of P to the pushdown stack and jump to the instruction located in memory location $\langle B_3 \rangle \langle B_2 \rangle$; otherwise, execute the next instruction in sequence.

In the above JUMP and CALL instructions $\langle B_2 \rangle$ contains the least significant half of the address and $\langle B_3 \rangle$ contains the most significant half of the address. Note that D_6 and D_7 of $\langle B_3 \rangle$ are "don't care" bits since the CPU uses fourteen bits of address.

RETURN INSTRUCTIONS – One Byte
(one cycle – PCI)
A return instruction may be used to exit from a subroutine; the stack is popped-up one level at a time.

RET	00 XXX 111	(P)←(Stack). Return to the instruction in the memory location addressed by the last value shifted into the pushdown stack. The stack pops up one level.
RFc (Return Condition False)	00 $0C_4C_3$ 011	If (c) = 0, (P)←(Stack); otherwise, (P) = (P)+1. If the content of flip-flop c is zero, then return to the instruction in the memory location addressed by the last value inserted in the pushdown stack. The stack pops up one level. Otherwise, execute the next instruction in sequence.
RTc (Return Condition True)	00 $1C_4C_3$ 011	If (c) = 1, (P)←(Stack); otherwise, (P) = (P)+1. If the content of flip-flop c is one, then return to the instruction in the memory location addressed by the last value inserted in the pushdown stack. The stack pops up one level. Otherwise, execute the next instruction in sequence.

RESTART INSTRUCTION – One Byte
(one cycle – PCI)
The restart instruction acts as a one byte call on eight specified locations of page 0, the first 256 instruction words.

RST	00 AAA 101	(Stack)←(P),(P)←(000000 00AAA000) Shift the contents of P to the pushdown stack. The content, AAA, of the instruction register is shifted into bits 3 through 5 of the P-counter. All other bits of the P-counter are set to zero. As a one-word "call", eight eight-byte subroutines may be accessed in the lower 64 words of memory.

INPUT/OUTPUT INSTRUCTIONS

One Byte
(two cycles – PCI/PCC)
Eight input devices may be referenced by the input instruction

INP	01 00M MM1	(A)←(input data lines). The content of register A is made available to external equipment at state T1 of the PCC cycle. The content of the instruction register is made available to external equipment at state T2 of the PCC cycle. New data for the accumulator is loaded at T3 of the PCC cycle. MMM denotes input device number. The content of the condition flip-flops, S,Z,P,C, is output on D_0, D_1, D_2, D_3 respectively at T4 on the PCC cycle.

Twenty-four output devices may be referenced by the output instruction.

OUT	01 RRM MM1	(Output data lines)←(A). The content of register A is made available to external equipment at state T1 and the content of the instruction register is made available to external equipment at state T2 of the PCC cycle. RRMMM denotes output device number (RR ≠ 00).

MACHINE INSTRUCTION

HALT INSTRUCTION – One Byte
(one cycle – PCI)

HLT	00 000 00X or 11 111 111	On receipt of the Halt Instruction, the activity of the processor is immediately suspended in the STOPPED state. The content of all registers and memory is unchanged. The P-counter has been updated and the internal dynamic memories continue to be refreshed.